University of Alberta
Printing Department

THE UNIVERSITY OF ALBERTA

THE GENERATION OF PSEUDO-RANDOM NUMBERS

by

David A. Scott

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

APRIL, 1967

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled THE GENERATION OF PSEUDO-RANDOM NUMBERS submitted by David A. Scott in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

The methods for generating and testing uniform pseudo-random numbers are reviewed. Some techniques for transforming uniform random deviates to non-uniform random deviates are described, principally for the normal distribution.

Applications show some uses to which sequences of pseudo-random numbers can be adapted. A simulation is introduced which examines probability estimates for a binomial distribution.

The Appendix shows results for some tests which indicate that multiplicative-congruential pseudo-random sequences are a satisfactory source of random numbers.

ACKNOWLEDGEMENTS

To Professor K.W. Smillie, I express my appreciation and thanks for his guidance in the preparation of this thesis. Also, I wish to thank Professors K.V. Wilson and U.M. von Maydell for their interest and assistance in this topic.

In addition, to Dr. D.B. Scott, Head of the Department of Computing Science, University of Alberta, my thanks for providing the facilities and assistance for carrying out this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Many investigations involving applied mathematics, statistics, or physics make use of Monte Carlo methods. For example, in applied mathematics, we may wish to obtain an approximation to the area under a curve, i.e., calculate $\int_0^1 f(x)dx$. We could select a pair of co-ordinates (x,y) at random and test the value of $f(x)$ against the value of y . If $f(x) \leq y$, then we would accept the point as falling on or below the curve; if $f(x) > y$, we would reject the point. Finally, we would compare the ratio of the number of accepted points to the total number of points selected at random. The ratio would represent an approximation to the area. Such a method would only give a very rough approximation to the area and would be used mainly in applications where a solution could not be obtained analytically. In physics, we could use Monte Carlo techniques to examine the behaviour of the diffusion of a gas, simulate random collisions of a molecule, or examine the shielding effects of a substance such as water. One very useful example is the simulation of the path of a particle with Brownian movement. In statistics, we might wish to simulate a game of chance, such as coin tossing or a queueing problem. In all the preceding examples,

Monte Carlo techniques would be used. Such techniques depend upon having available sequences of numbers which appear to have been drawn at random from a particular probability distribution.

Monte Carlo techniques are able to give at least approximate answers where other techniques fail. For example, an experiment to examine absorption of X-rays would be very difficult to control and to obtain any results from. However, with Monte Carlo methods, we would simply generate a sequence of random numbers to follow the history of an individual ray, perform this experiment for a sufficiently large number of trials and tabulate the results.

It is the aim of this thesis to examine sequences of random numbers generated by digital computer programs and to report results pertaining to some special sequences.

Chapter II examines the methods which have been used in the past and current methods for generating uniformly-distributed sequences of random numbers. It shows the advantages and disadvantages of several methods for generating random sequences on computers from the point of view of statistics and the speed with which the sequences are generated.

Chapter III surveys different statistical tests for

examining sequences of numbers for randomness. Both methods used in the past and current methods are examined. The emphasis is placed on four tests devised by Kendall and Babington-Smith.

Chapter IV considers the problem of transforming a uniformly distributed random variable into, particularly, a normally distributed random variable. Five different transformations are illustrated and their advantages and disadvantages are discussed. The transformation to an exponentially distributed random variable and a Poisson variable are treated briefly.

Chapter V discusses four simple applications, three from probability theory, and one from mathematics.

CHAPTER II

A SURVEY OF METHODS FOR GENERATING RANDOM NUMBERS

2.1 Introduction

Student (1908) appears to have been the first to use random sampling techniques to estimate distribution functions. His method of choosing a random number consisted of using a correlation table of the heights and the left middle finger measurements of 3000 criminals from a paper by MacDonell (1901). The digits from the table were written on 3000 pieces of cardboard, shuffled, and were drawn at random, with replacement, four at a time. This method, however, proved to be very slow, and it was very difficult to determine when the pieces of cardboard had been shuffled well. Karl Pearson suggested to Tippett (1925) that he should replace the entire system of ticket sampling by a table of random numbers ranging from 0000 to 9999. Tippett's table (1927) was formed by taking 40,000 digits "at random" from census tables. They were grouped in fours to give the required 10,000 numbers. Karl Pearson has shown in the foreword how these numbers can be converted to give random samples from a non-uniform distribution. Fisher and Yates (1938) produced a table of 15,000 random sampling numbers. Their table was

compiled from among the fifteenth-to-nineteenth digits in certain sections of Thompson's "Logarithmica Brittanica". Fisher and Yates applied the frequency test to their table of 15,000 digits. They found an excess of the digits 3, 6, and 9 as shown in Table 2.1.1. If the digits were uniformly distributed, then the expected frequency of each digit would be 1500.

Digit	Frequency
0	1493
1	1441
2	1461
3	* 1552
4	1494
5	1454
6	* 1613
7	1491
8	1482
9	* 1519

Table 2.1.1 - Observed Frequencies of Fisher and Yates Digits

In order to correct the table, they removed fifty of the sixes "strictly at random" and replaced them by one of the other nine digits "strictly at random". The tables

then satisfied the tests applied by Fisher and Yates.

2.2 Mechanical Methods for Generating Random Numbers

Kendall and Babington-Smith (1938,1939a) had considered forming random sampling numbers from mathematical tables but decided against it in favour of a mechanical method. There were known to be non-random properties in sequences of digits from mathematical tables, such as tables of logarithms. As an example, Kendall and Babington-Smith cite the following theorem proven by Franel (1917):

Theorem 2.2.1: Consider the logarithms to base 10 of the natural numbers from 1 onwards. The proportional frequency of any digit in this series does not tend to a limit.

Such a series of numbers will contain an increasing number of runs of certain digits.

Kendall and Babington-Smith were the first to successfully generate random numbers by means of a mechanical device. Previously, mechanical methods had been considered untrustworthy. However, Kendall and Babington-Smith's machine was designed to eliminate the sources of bias which had appeared in other generators. Their randomizing machine was composed of a disc which was rotated by an electric motor at a very rapid rate in a darkened room. The disc was divided equally into ten

sectors on which appeared the digits 0 through 9, inclusive. An electric spark or a neon lamp illuminated the disc instantaneously so as to make the disc appear stationary for a moment. When the flash occurred, a number was selected from the disc by means of a fixed pointer. The intervals of the flashes were varied by means of a neon lamp in parallel with a condensor in an independent electrical circuit. To add to the randomness, a key tapped by an observer broke the circuit intermittantly at irregular intervals. The table of random numbers generated by this mechanism (see Kendall and Babington-Smith (1939b)) satisfied the requirements for random numbers for approximately ten years.

In addition to constructing a table of random numbers, Kendall and Babington-Smith devised a series of tests for sequences of random numbers. These tests will be described in the next chapter.

The most extensive table of random digits, to date, has been published by the RAND Corporation (1955). The random digits were produced by a randomization of a basic table generated by an electronic roulette wheel. The process required a frequency pulse source providing, on the average, about 100,000 pulses per second. These pulses were gated approximately once per second by a constant frequency pulse. Pulse standardization circuits

passed the pulses through a five-place binary counter. In principle, the machine was a thirty-two place roulette wheel which made approximately 3000 revolutions per trial and produced one number per second. A binary-to-decimal converter was used to convert twenty of the thirty-two digits produced and only the final digit of the two-digit decimal number was retained. The final digit was then punched on a card. Production of the digits began in April of 1947. After 500,000 digits were produced, tests were performed on them with satisfactory results. Later, after continuous running of the generator for more than a month, tests showed that there seemed to be a slight tendency favouring even digits more than odd digits. From this, it was concluded that the machine had probably been running down during the month. In order to correct the fault, it was decided to randomize the digits produced for the table. Each of the fifty digits punched on a card was added, modulo 10, to the corresponding digit of the previous card and the result punched on another card. Tests performed on the derived series yielded acceptable results and were adopted as the table of random digits which was published.

2.3 Random Number Generation with an Automatic Computer

With the introduction of computers during the late

1940's, the use of tables of random digits became almost impossible and was certainly undesirable. The tables required large amounts of space for storage, and access to auxiliary storage devices by the computer was slow. In order to be able to use random digits, it was necessary to derive some method of generating them within the computer itself. Two types of generators, physical and arithmetic, have been suggested for internal computer use, and will be discussed in the remaining sections of this chapter. The emphasis will be on arithmetic generators.

2.4 Physical Random Number Generators

Basically, a physical random number generator consists of some external device which delivers a series of random pulses to the computer. The computer, in turn, transforms the series of pulses into a random number. There are two main methods of obtaining random numbers by means of a physical process. The first method is based on the noise of electronic tubes, and the second on the radiation of radioactive substances.

The basis of the first method is a noise generator. In electronic circuits, there is inherent fluctuating noise, which, with appropriate amplificaton, will ensure a fluctuation in the output voltage. One particular noise generator circuit described by Shreider (1964) uses

a gas-discharge tube and a magnet. Noise pulses can be obtained directly from the gas-discharge tube with a suitable orientation of the magnet. The output signals from the tube are applied to the input of an amplifier.

Radioactive sources for random number generation usually consist of a radiation source of radioactive particles plus a counter. The counter registers the number of radioactive particles released in an interval of time Δt . If the number of particles is even, the digit zero is recorded; if the number of particles is odd, the digit one is recorded as the value of the random digit.

Pawlak (1956) has used a random number generator based on the random state of an electronic circuit. He lets A and B represent two possible states of a flip-flop. If the contact is switched on, the flip-flop will be randomly at one of its states A and B . From this base, we may get a sequence of 2k random elements. This will give us a random sequence of states A and B which are statistically independent. One of the sets of positions of the flip-flop after switching may be:

ABBAAABBABBBAABAB.

One quite well known physical random number generator is ERNIE (Electronic Random Number Indicator Equipment) described by Thompson (1959). The machine is used in the British General Post Office's Premium Savings Bond scheme. In the scheme, certain bond numbers are eligible to take part in a draw, and, if the number generated by ERNIE coincides with one of the bond numbers, a prize is awarded to the owner of that bond number. In the first stage, ERNIE generates and prints a sample, with replacement, from the population of eligible bond numbers. In the second stage, reference is made to the records, and, if a bond number which appears has already been awarded a better prize than the current one, the number is deleted from the list, i.e., only one prize can be won at a time. Each bond number contains nine digits and therefore ERNIE must generate nine random digits. In order to do this, the electrical noise in ten neon tubes is used. The bond numbers produced in any run depend on the particular noise waveforms produced by the ten neon tubes. Figure 2.4.1 shows the parts of a digit generator. Each time the noise waveform passes upwardly through a certain fixed level, the pulse generator emits a short pulse of approximately two microseconds. After each pulse is generated, there is a delay time of thirty microseconds

before another pulse can be generated by the noise. A train of pulses is thus passed through an n-position cyclic counter, each pulse advancing the counter by one. The counter is examined at regular intervals, and the digit indicated is passed to the output as one of a sequence of random digits. In order that no fault should develop during the selection of bond numbers, a redundancy technique has been incorporated in the generation of random digits. Ten cyclic counters are provided to give nine random digits. The counters are

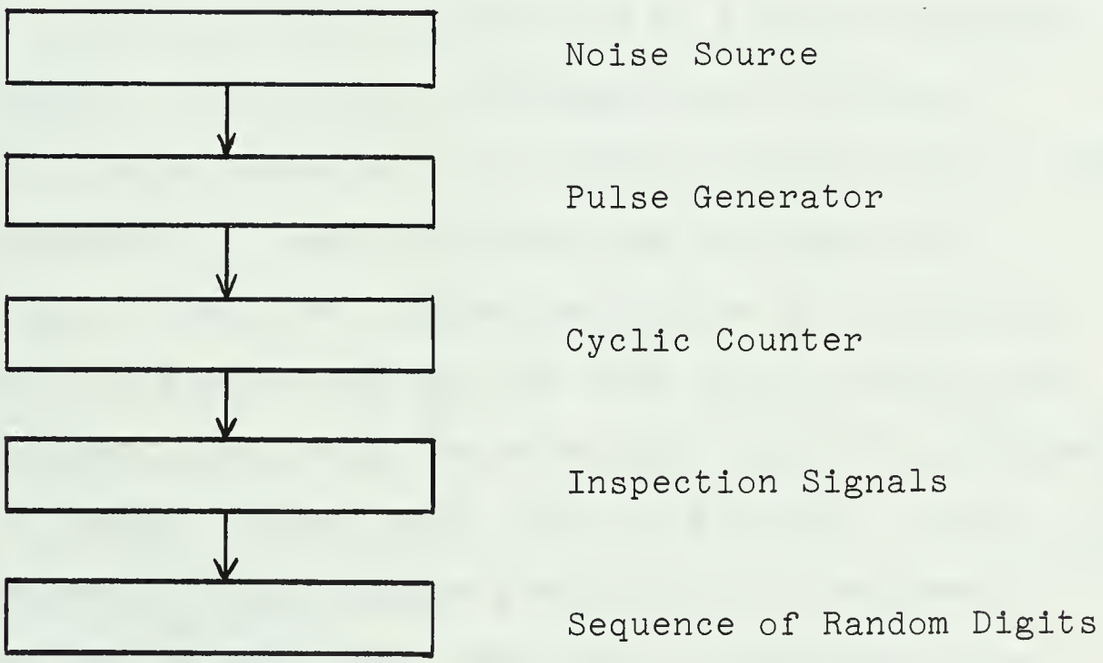


Figure 2.4.1 - ERNIE Digit Generator

connected in pairs and the final outputs are taken as the differences of each pair of generators. It might be pointed out that statistical tests performed on ERNIE satisfied the criteria for randomness. However, the use of ERNIE as a general purpose generator is very restrictive since the speed of generation is extremely slow. For the application it was designed, ERNIE is excellent because relatively few bond numbers are required at a time.

2.5 Arithmetic Random Number Generators

Arithmetic generators all have the same basic property - two numbers are multiplied together and some part of their product is retained as a random sequence of digits. The distinct advantage with arithmetic generators as compared to the physical generators is that any sequence of digits produced can be reproduced exactly, if required. This enables one to correct an error in a program and use the same data to check the correction, i.e., only the correction and not a different set of random numbers will affect the result. Since all sequences of random digits produced by an arithmetic method can be predicted, they will be referred to as pseudo-random numbers. They will be "random" so far as satisfying the criteria for randomness discussed in the next chapter is concerned.

The first arithmetic generator was proposed in 1946 by von Neumann and Metropolis (see Richtmeyer (1961)). This was the middle-of-the-squares process. An n-digit number x_0 is squared to give a 2n-digit product x_0^2. The middle n digits are retained as x_1 and the process is repeated. As an example, consider $x_0 = 1234$, then $x_0^2 = 01522756$ and $x_1 = 5227$. $x_1^2 = 27321529$ and therefore $x_2 = 3215$. Tests performed on the numbers generated by the middle-of-the-squares method produced unsatisfactory results. The length of a cycle is dependent on the starting value and some initial values can lead to short cycles. For example, with $x_0 = 3600$, the sequence 9600, 1600, 5600, 3600, 9600,... is produced. Cycles of this type are difficult to detect as they may occur at any point during the generation of the sequences. Another drawback of the middle-of-the-squares is that certain values can lead to a degenerate cycle, i.e., a result where the length of the cycle is one. Apart from the preceding problems, the middle-of-the-squares generator does not give a uniform distribution of digits regardless of the starting value.

An improvement to the middle-of-the-squares generator is the mid-product method (see Forsythe (1951)). Initially, two n-digit values are needed. The two values

are multiplied together to form the succeeding number and the process is repeated. For example $x_0 = 1234$, $x_1 = 5678$ give the series 0066, 3747, 2473, 2663,... . The advantage of the mid-product generator as compared to the middle-of-the-squares generator is that a much longer cycle may be obtained. Both initial values must occur consecutively before the cycle can repeat. However, this method, too, has values which can lead to a degenerate cycle of unit length. For example, with $x_0 = 0100$, $x_1 = 0500$, the series is 0500, 2500, 2500, 2500,... .

Due to the finite word-length of a computer, any sequence of random digits must eventually repeat itself. The problem, then, is to make the cycle as long as possible. A solution was proposed by Lehmer (1949). His method, to date, has been the most successful and widely-used for generating pseudo-random numbers on a computer. It eliminates any possibility of a sequence degenerating into a small loop.

In order to be able to understand Lehmer's method, we must first consider the definition of primitive roots as given, for example, in Abramowitz and Stegun (1964).

Definition 2.5.1: The integers not exceeding and relatively prime to a fixed integer n form a group; the

group is cyclic if and only if $n = 2$ or 4 or n is of the form p^k or $2p^k$, where p is an odd prime. Then g is a primitive root of n if it generates that group, i.e., if $g, g^2, g^3, \ldots, g^{\Phi(n)}$ are distinct modulo n. There are $\Phi(\Phi(n))$ primitive roots of n, where $\Phi(n)$ is the number of integers not exceeding and relatively prime to n (Euler's Φ function).

The results for selected integers are shown in Table 2.5.1.

n	$\Phi(n)$	$\Phi(\Phi(n))$	Relatively Prime Integers	Primitive Roots of n
11	10	4	1 2 3 4 5 6 7 8 9 10	2 6 7 8
9	6	2	1 2 4 5 7 8	2 5
18	6	2	1 5 7 11 13 17	5 11
10	4	2	1 3 7 9	3 7
15	8	4	1 2 4 7 8 11 13 14	NONE

Table 2.5.1 - Primitive Roots of Selected Integers

For the derivation of a method for finding primitive roots of primes, consult Ore (1948). For powers of 2, the primitive roots are $r = 1$, and $r = 3 \pmod 4$. For modulo 8 and higher powers of 2, no primitive roots exist because all odd numbers are relatively prime to the moduli and have the form

$$a = 2n + 1 .$$

Therefore

$$a^2 = 4n^2 + 4n + 1 = 4(n+1)n + 1 .$$

One of n or $n + 1$ must contain a factor of 2, so that

$$a^2 \equiv 1 \pmod 8 .$$

But $\Phi(8) = 4$ and therefore has no primitive roots. Since

$$a^2 = 1 + 8t ,$$

then

$$a^4 \equiv 1 \pmod{16} ,$$

$$a^8 \equiv 1 \pmod{32} ,$$

and

$$a^{2\alpha-2} \equiv 1 \pmod{2^\alpha} . \qquad (2.5.1)$$

Since $\Phi(2^\alpha) = 2^{\alpha-1}$, (2.5.1) shows that there can be no primitive roots for powers of 2 higher than $\alpha = 2$. Equation (2.5.1) also implies that the highest exponent to which a number can possibly belong (mod 2^α) is $2^{\alpha-2}$ if $\alpha \geq 3$.

By a theorem in number theory, among the powers of 2, only 2 and 4 have primitive roots. For all higher powers of 2^{α} , $\alpha \geq 3$, every odd number satisfies the congruence

$$a^{2^{\alpha-2}} = a^{\frac{1}{2}\Phi(2^{\alpha})} \equiv 1 \pmod{2^{\alpha}} \qquad (2.5.2)$$

Lehmer's method, or the multiplicative-congruential method, forms a sequence of pseudo-random numbers according to the formula

$$x_{i+1} \equiv a\, x_i \pmod{M} \qquad (2.5.3)$$

for given a and x_0 subject to certain restrictions which must be observed to ensure a maximal period. These restrictions may be summarized in the following theorem:

Theorem 2.5.1: The sequence defined by the congruence relation (2.5.3) has maximal period provided that

i) x_0 is relatively prime to M ,

ii) a is a primitive root for p^{α} , if p^{α} is a factor of M , with p odd and α as large as possible.

or, with $p = 2$ and $\alpha = 1$ or 2 .

iii) a belongs to $2^{\alpha-2}$ if 2^{α} is a factor of M , with $\alpha \geq 2$. Moreover, for any value of M , there exist values of a satisfying these conditions, and, finally, the maximal period is the lowest common multiple of the periods, $(p-1)p^{\alpha-1}$ or $2^{\alpha-2}$ with respect to the prime power factors.

It is relatively easy to satisfy condition (i). Condition (iii) may be satisfied if $a \equiv \pm 3 \pmod 8$. To show that this is true, let us follow an example with $M = 2^p$. In this case, a must be odd to be relatively prime to 2^p , i.e., $a = 2m+1$. Repeated application of formula 2.5.3 gives

$$x_{i+n} \equiv a^n x_i \pmod{2^p} \qquad (2.5.4)$$

The period of the sequence is defined to be the smallest integer n_0 for which $a^n \equiv 1 \pmod{2^p}$ and we have already seen that n_0 for 2^p is 2^{p-2} . The general form of a^n is

$$a^n = (1+2m)^n = 1+n\ 2m+\binom{n}{2}(2m)^2+\binom{n}{3}(2m)^3+\binom{n}{4}(2m)^4+\ldots \qquad (2.5.5)$$

For $n = 2^{p-2}$, (2.5.5) becomes

$$a^{2^{p-2}} = 1+2^{p-2}2m+\frac{2^{p-2}(2^{p-2}-1)}{2!}(2m)^2+\frac{2^{p-2}(2^{p-2}-1)(2^{p-2}-2)}{3!}(2m)^3$$

$$+ \text{ higher powers of at least } 2^p$$

$$a^{2^{p-2}} \equiv 1+2^{p-1}m-2^{p-1}m^2 \pmod{2^p}$$

$$\equiv 1+2^{p-1}m(1-m) \pmod{2^p} . \qquad (2.5.6)$$

If m is odd, let $m = 2b+1$. Then (2.5.6) becomes

$$a^{2^{p-2}} \equiv 1+2^{p-1}(2b+1)(1-(2b+1)) \pmod{2^p}$$

$$\equiv 1 \pmod{2^p} .$$

If m is even, let $m = 2b$. Then (2.5.6) becomes

$$a^{2^{p-2}} \equiv 1+2^{p-1}(2b)(1-(2b)) \pmod{2^p}$$

$$\equiv 1 \pmod{2^p}.$$

Thus $a^{2^{p-2}} \equiv 1 \pmod{2^p}$ for all a .

For $n = 2^{p-3}$, (2.5.5) becomes

$$a^{2^{p-3}} = 1+2^{p-2}m+\frac{2^{p-2}(2^{p-2}-1)}{2!}m^2+\frac{2^{p-3}(2^{p-3}-1)(2^{p-3}-2)}{3!}(2m)^3$$

$$+\frac{2^{p-3}(2^{p-3}-1)(2^{p-3}-2)(2^{p-3}-3)}{4!}(2m)^4$$

$$+ \text{ higher powers of } 2 \text{ than } 2^p$$

$$\equiv 1+2^{p-2}m-2^{p-2}m^2-2^{p-1}m^4 \pmod{2^p} . \qquad (2.5.7)$$

If m is even, i.e., if $m = 2b$, (2.5.7) becomes

$$a^{2^{p-3}} \equiv 1+2^{p-1}b \pmod{2^p} .$$

Then if b is odd,

$$a^{2^{p-3}} \equiv 1+2^{p-1} \pmod{2^p}$$

or, if b is even,

$$a^{2^{p-3}} \equiv 1 \pmod{2^p} .$$

If m is odd, i.e., $m = 2b+1$, (2.5.7) becomes

$$a^{2^{p-3}} \equiv 1+2^{p-2}(2b+1)-2^{p-2}(2b+1)^2-2^{p-1}(2b+1)^4 \pmod{2^p}$$

$$\equiv 1+2^{p-1}b+2^{p-2}-2^{p-2}-2^{p-1} \pmod{2^p}$$

$$\equiv 1+2^{p-1}(b-1) \pmod{2^p} .$$

Then if b is odd,

$$a^{2^{p-3}} \equiv 1+2^{p-1}(2c+1-1) \pmod{2^p}$$

$$\equiv 1 \pmod{2^p}$$

or, if b is even,

$$a^{2^{p-3}} \equiv 1+2^{p-1}(2c-1) \pmod{2^p}$$

$$\equiv 1+2^{p-1} \pmod{2^p} .$$

From the preceding, it can be readily seen that

$$a \equiv \pm 3 \pmod{8} .$$

Examples of multipliers and their resulting sequences will be shown in the next chapter.

A variation to Lehmer's method consists of adding a constant c to equation (2.5.3) to give

$$x_{i+1} \equiv a\, x_i + c \pmod{M} \qquad (2.5.8)$$

This variation is referred to as the mixed-congruential method. In order to ensure the maximum period, conditions upon a, x, and c may be stated in the following theorem:

Theorem 2.5.2: The sequence defined by (2.5.8) has full period M, provided that

i) c is relatively prime to M,

ii) $a \equiv 1 \pmod{p}$ if p is a prime factor of M,

iii) $a \equiv 1 \pmod{4}$ if 4 is a factor of M.

The proof of Theorem 2.5.1 may be found in Hull and Dobell (1962).

The mixed-congruential generators have a few advantages and some disadvantages when compared to the multiplicative-congruential generators. The main advantage of the mixed generators lies in the fact that a full period M can be attained whereas it cannot be attained with multiplicative generators. Common multipliers for the mixed methods were $a = 2^p+1$ for binary machines or $a = 10^p+1$ for decimal machines. The value of p is the number of binary bit positions or the number of decimal digit positions available to represent an integer in the computer being used. This enabled a simple shift and add rather than a full multiplication to take place. However, it was multipliers of this type which gave rise to rather poor statistical behaviour (see Hull and Dobell (1964)). With multiplicative-congruential generators, the statistical behaviour of sequences satisfying the conditions of Theorem 2.5.1

is generally acceptable.

Another type of pseudo-random number generator which has been proposed is an additive type. The principal reason for introducing such a generator is its speed over the multiplicative methods since addition is performed more quickly than multiplication in most computers. One additive generator which has been suggested (see Taussky and Todd (1956)) is one involving a reduced Fibonacci sequence. Initially, $F_0 = 0$ and $F_1 = 1$, succeeding numbers are generated by

$$F_{n+2} \equiv F_{n+1}+F_n \pmod{M} \qquad (2.5.9)$$

The results of this generator showed that the speed of generation and the period of these numbers were satisfactory, but that the successive members of F_n were not independent (see Taussky and Todd (1956)). As a remedy, alternate members of the sequence were chosen. The results showed a more satisfactory behaviour, but were still not as good as with the multiplicative generators. Green, Smith and Klem (1959) suggested a different method for an additive pseudo-random number generator.

Their method used the formulae

$$X_j \equiv (X_{j-1}+X_{j-n}) \bmod 1$$

for decimal machines, and

$$X_j \equiv (X_{j-1}+X_{j-n}) \bmod 2^p$$

for binary machines. In this method, the most recent n random numbers must be stored. The cycle will, of course, be periodic as soon as the original n numbers are generated. Tests performed by Green, Smith and Klem showed the generator to be unsatisfactory. To remedy this situation, alternate members were chosen. The results from the multiplicative-congruential generators, however, were still more acceptable.

CHAPTER III

RANDOM NUMBER TESTING

3.1 Introduction

The purpose of this chapter is to examine various techniques for testing a sequence of digits or a sequence of numbers for random properties. First, four basic tests due to Kendall and Babington-Smith will be examined, then more recent tests specific to computer-generated pseudo-random sequences will be discussed. Some results from the tests are given in the Appendix for the multiplicative-congruential method.

We should note that the sequences of pseudo-random numbers were generated and tested on an IBM 7040 computer, a thirty-six bit binary machine. For this reason, most of the chapter will be restricted to the consideration of sequences of octal digits. It is a relatively simple task to convert the test to consider the decimal case.

3.2 Four Basic Tests for Random Digits

In addition to constructing a mechanical random digit generator (see 2.2), Kendall and Babington-Smith (1938,1939a) also derived four basic tests for randomness in a sequence of digits. The tests were the frequency test, the serial test, the poker test, and the gap test. Each of these tests will be discussed in turn.

The frequency test is probably the easiest to understand. In a sequence of random octal digits, we would expect to find every digit occurring approximately an equal number of times. For example, we would expect the octal digit 5 to occur 12.5 percent of the time in a set of random octal digits. Any marked departure from equality of frequencies would lead one to suspect a bias toward some digits and away from others. Kendall and Babington-Smith used a chi-square test to test the hypothesis of uniformity, i.e., equal frequencies for each digit.

The serial test involves a similar type of test with pairs of digits. We would expect that no single digit should tend to precede or follow another digit if a series of digits were to be locally random. In order to use the test, a two-way table would be constructed. The entries in the table would be dependent on the sequence of digits examined. For example, if an octal digit 4 followed an octal digit 6 in the sequence, then we would place an entry in the fourth row and sixth column of the table. When the entire sequence of digits has been examined, the number of entries in each element of the table would be counted and we would expect the totals to be approximately equal. This hypothesis could be tested using a chi-square test with sixty-three degrees

of freedom.

As an example of the two-way table for serial frequencies, consider the sequence of digits

15101332462566021115 2 .

Then, Table 3.2.1 shows the serial frequencies of the digits.

	0	1	2	3	4	5	6	7
0	0	1	1	0	0	0	0	0
1	1	2	0	1	0	2	0	0
2	0	2	0	0	1	1	0	0
3	0	0	1	1	0	0	0	0
4	0	0	0	0	0	0	1	0
5	0	1	1	0	0	0	1	0
6	1	0	1	0	0	0	1	0
7	0	0	0	0	0	0	0	0

Table 3.2.1 - Serial Frequency of Some Octal Digits

A generalization of the serial test, not suggested by Kendall and Babington-Smith, would be to compare two digits which are separated from each other by n digits, where n is any positive integer. We would expect the results from this modification to be exactly the same for any value of n .

The poker test examines groups of five digits and tests their value as a poker hand. The events and their expected occurrance are shown in Table 3.2.2.

Event	Description	Example	Probability
Bust	All digits different	abcde	.205
1 Pair	Two digits the same	aabcd	.513
2 Pairs	Two pairs of digits the same	aabbc	.153
Triple	Three digits the same	aaabc	.103
Full House	One pair, one triple	aaabb	.017
4 of a Kind	Four digits the same	aaaab	.009
5 of a Kind	All digits the same	aaaaa	.000
			1.000

Table 3.2.2 - Probabilities for Selected Poker Hands

In order to use the poker test, the sequence of digits is divided into groups of five digits and the observed frequency of the possible poker hands is compared to the theoretical frequency by a chi-square test. Since the probabilities of a full house, four of a kind and five of a kind are relatively small, the frequencies of these events are usually grouped together.

The gap test compares the expectations with the

frequency of the gaps between two successive equal digits. A gap may be defined as follows. If a digit n is separated from the same digit n by m digits, this constitutes a gap of length m for the digit n . The probabilities for octal digit gaps may be summarized in Table 3.2.3.

Length of Gap	Probability
0	.125
1	.109
2	.095
3	.083
4	.073
5	.064
6	.056
7	.049
8	.043
9	.038
10	.033
11	.029
12	.025
13	.022
14	.019
15	.017
16-20	.058
21-25	.030
$\geq$26	.032
	1.000

Table 3.2.3 - Gap Probabilities for Octal Digits

For the gap test, we may test all eight octal digits. However, usually it is sufficient to test, for example, the gaps for the octal digit 2 since we would expect similar results for any other octal digit.

Kendall and Babington-Smith were careful to caution that if two locally random sets of digits, i.e., sets whose digits pass the preceding four tests, were combined, the resulting sequence of digits would not necessarily be locally random. Furthermore, as the number of random digits in the set increased, the number of bad patches appearing without local randomness was bound to increase.

The chi-square test was used by Kendall and Babington-Smith to measure permissable deviation from expectations. The table of random digits produced by the two authors satisfied the four tests and was widely used until the production of RAND's million random digits.

3.3 Modifications to Kendall and Babington-Smith's Tests

The frequency test can be applied in two different ways. First, a count can be made of the frequency of occurrence of each octal digit in the set of pseudo-random numbers. Second, a count can be made of each digit position of the pseudo-random number. For example, let us consider the ten sequences of octal digits in Table

3.3.1 which were generated by the relation

$x_{i+1} \equiv (2^{18}+3) \pmod{2^{35}}$, where $x_0 = (123456789)_{10}$.

366017263477

073715432675

101107120467

353560761645

274045325357

236747600315

235035601147

227614603465

211100612637

140621640335

Table 3.3.1 - First Ten Numbers from

$$x_{i+1} \equiv (2^{18}+3)x_i \pmod{2^{35}}$$

$$x_0 \equiv (123456789)_{10}$$

The first frequency table is shown in Table 3.3.2.

Digit	Frequency
0	16
1	17
2	12
3	16
4	12
5	13
6	18
7	16
	120

Table 3.3.2 - Observed Frequencies of Octal Digits of Table 3.3.1

The frequency of the octal digits would then be compared to the expected frequencies of fifteen by a chi-square test with seven degrees of freedom.

The second frequency test mentioned above would be tallied in a similar fashion except that there would be twelve separate tables each with eight entries. An example of this type of table is illustrated in Table 3.3.3

Digit	Octal Position											
	0	1	2	3	4	5	6	7	8	9	10	11
0	1	1	1	3	2	2	0	3	3	0	0	0
1	2	1	2	2	3	1	1	1	2	1	1	0
2	5	1	0	0	1	0	1	2	2	0	0	0
3	2	2	2	0	1	0	1	1	2	3	2	0
4	0	1	1	0	2	1	1	1	0	3	2	0
5	0	1	1	1	0	3	0	0	1	0	1	5
6	0	1	2	2	1	0	5	2	0	3	2	0
7	0	2	1	2	0	3	1	0	0	0	2	5

Table 3.3.3 - Observed Frequencies for Individual Octal Digits from Table 3.3.1

Of course, one can see immediately from inspection of Table 3.3.1 that the last digit will always be odd. This is a result of the restrictions on x_0 and a in the multiplicative congruential method. The first digits appearing in Table 3.3.1 are always 0, 1, 2, or 3 because of the octal representation of numbers in the 7040 - the first octal digit always contains the sign of the numbers. These phenomena force us to neglect some of the octal digit positions in a pseudo-random number.

A further variation to the second method is to divide the interval (0,1) into equal subintervals and to test the observed frequency of numbers within the interval

with the expected frequency. However, the number of intervals, usually restricts the test to the first few octal digits. For example, if we divide the interval into eight equal segments, then we would only be testing the first octal digit.

The serial test for pseudo-random numbers can be performed in a manner similar to that of the frequency test. Let us consider the sequence of numbers generated in Table 3.3.1. Then the serial test performed on the entire set of numbers would give the results shown in Table 3.3.4.

	0	1	2	3	4	5	6	7
0	2	3	0	4	2	0	2	3
1	3	4	2	0	3	2	2	1
2	1	2	1	2	0	1	3	2
3	0	1	2	1	2	6	2	2
4	3	0	0	1	0	2	3	3
5	1	1	3	3	1	0	2	1
6	5	3	1	2	2	1	1	3
7	1	3	3	2	2	1	3	1

Table 3.3.4 - Serial Frequencies for the Octal Digits in Table 3.3.1

The observed frequencies are compared to the expected

frequencies by means of the chi-square test.

A second method for calculating serial frequency would be to examine each octal digit position separately in a manner similar to that of the frequency test.

Hull and Dobell (1964) referred to a paper by Good (1953) in which the results from the frequency test and the serial test were combined to test a sequence for randomness. It was shown in the latter paper that if χ_1^2 represents the chi-square from the frequency test and χ_2^2 represents the chi-square from the serial test, then $\chi_D^2 = \chi_2^2 - \chi_1^2$, is asymptotically distributed as a chi-square distribution with $\nu^2 - \nu$ degrees of freedom where $\nu = 8$ for the case of octal numbers. If we are examining the frequency of numbers within equal subintervals, then ν is the number of intervals.

The poker test can also be formulated in two ways. First, we could isolate the digits of a pseudo-random number serially, five at a time, and test the observed frequency to the expected frequency. Second, we could examine the individual octal positions in the same manner as in the frequency test and perform a similar chi-square test.

The gap test could also be treated in the same manner, but its use seems to have been emphasized less in the

current literature than the use of the serial and frequency tests.

A further application of the chi-square test from the values of chi-square obtained from Kendall and Babington-Smith tests was suggested by Green, Smith, and Klem (1959). When the values of chi-square are calculated for a sufficiently large number of frequency and serial tests, we should obtain a whole range of chi-square values. The frequency of these values should in turn, also, satisfy a chi-square distribution with n intervals between 0 and 100 percent. This additional test has not been used to a wide extent since it requires the generation of several blocks of pseudo-random numbers and the testing of each individual block.

3.4 Other Tests for Randomness

Although Kendall and Babington-Smith's tests still form the basis for testing pseudo-random numbers, other tests have been suggested in recent years.

Gruenberger and Mark (1951) suggested a test for use in Monte Carlo calculations. Since many Monte Carlo techniques require the use of special coordinates, two successive pseudo-random numbers can be used to determine (x,y)-coordinates within the unit square. The test

suggested an examination of the square of the distance between two successive points. For two points, the probability that the square of the distance between them is α^2 is given by

$$P = \pi\alpha^2 - \frac{8\alpha^3}{3} + \frac{\alpha^4}{2} \qquad \text{for } 0 \leq \alpha^2 < 1.0$$

and

$$P = \frac{1}{3} + (\pi-2)\alpha^2 + 4(\alpha^2-1)^{1/2} + \frac{8}{3}(\alpha^2-1)^{3/2} - \frac{\alpha^4}{2} - 4\alpha^2\sec^{-1}\alpha$$

$$\text{for } 1.0 \leq \alpha^2 \leq 2.0 \qquad (3.4.1)$$

In order to perform the test, the square of the distance is computed and the distribution of the results is compared to the theoretical by the chi-square test. The cumulative probabilities for discreet α^2 are shown in Table 3.4.1 (see Gruenberger and Mark (1951)) as well as the probabilities for each of the intervals 0.0 (0.1) 2.0.

α^2	$P((\alpha-0.1)^2<d^2\leq\alpha^2)$	Cumulative Probability
0.1	.234832	.234832
0.2	.174973	.409805
0.3	.139495	.549300
0.4	.112718	.662018
0.5	.090971	.752987
0.6	.072614	.825601
0.7	.056748	.882349
0.8	.042814	.925163
0.9	.030430	.955593
1.0	.019333	.974926
1.1	.010777	.985703
1.2	.006345	.992048
1.3	.003740	.995788
1.4	.002138	.997926
1.5	.001154	.999080
1.6	.000572	.999652
1.7	.000246	.999898
1.8	.000084	.999982
1.9	.000017	.999999
2.0	.000001	1.000000

Table 3.4.1 - Probabilities Associated with the d^2-Test

Gorenstein (1967) described tests on the moments of the uniform distribution. He computed the mean and the second and third moments for equal blocks of generated pseudo-random sequences. The expected values are shown in Table 3.4.2 for a uniform generator with modulus 2^p.

Moment	Distribution	Expected Value
First	$\frac{1}{2^p N} \sum_{i=1}^{N} x_i$	1/2
Second	$\frac{1}{2^p 2^p N} \sum_{i=1}^{N} x_1^2$	1/3
Third	$\frac{1}{2^p 2^p 2^p N} \sum_{i=1}^{N} x_1^3$	1/4

Table 3.4.2 - Moments for Gorenstein's Tests

McLaren and Marsaglia (1965) used two simple applications of order statistics to perform additional tests for pseudo-random sequences. Their tests were based on the following properties. From a set of n pseudo-random numbers, a maximum element, Max, and a

minimum element, Min, are chosen. If the n pseudo-random numbers $\{x_1,\ldots,x_n\}$ are uniformly distributed, then Max should have a distribution $P(\text{Max} < a) = F(a) = a^n$ for $0 < a < 1$. Similarly, Min should have a distribution $P(\text{Min} > a) = \Phi(a) = (1-a)^n$. The chi-square test is again used to test equal subintervals of a large block of pseudo-random numbers.

3.5 Additional Comments

Peach (1961) discovered an interesting phenomenon about the mixed congruential methods. He considered the example $x_{n+1} \equiv 9x_n + 13 \pmod{32}$ which generated the sequence

0	13	2	31	4	17	6	3	
8	21	10	7	12	25	14	11	
16	29	18	15	20	1	22	19	
24	5	26	23	28	9	30	27	0

Each of the above rows comprises a quarter-period of the sequence. Upon close examination of the half-periods, it can be seen that each corresponding number differs from the other by exactly sixteen. Similarly, for each quarter-period, the difference is eight, and for each eighth-period the difference is four. In general, it was observed that half-periods differed by 2^{m-1}, quarter-

periods differed by 2^{m-2}, eighth-periods differed by 2^{m-3}, etc. These periodicities lead us to caution that pseudo-random numbers generated by the mixed-congruential method contain patterns and periodicities which act as constraints upon their variability.

With the preceding example in mind, let us now examine a multiplicative-congruential generator, specifically $x_{i+1} = 29x_i \pmod{64}$ with $x_0 = 11$. The sequence generated is

11 63 35 55 59 47 19 39
43 31 3 23 27 15 51 7 11

Here, we have the same phenomenon as before - half-sequences differ by thirty-two, quarter-sequences differ by sixteen.

Greenberger (1965) discovered that the multiplicative congruential generator $x_{i+1} \equiv (2^{18}+3)x_i \pmod{2^{35}}$ produced a second-order serial correlation. In order to rectify the situation, a series of five successive multipliers close to, but not equal to, $2^{18} + 3$ were used, the successive multipliers being chosen "at random".

In the multiplicative-congruential method, only half of the odd numbers can possibly occur. For example, if $a \equiv 3 \pmod 8$ and $x_0 \equiv 1 \pmod 8$, then $x_{i+1} \equiv 1 \pmod 8$ or $x_{i+1} \equiv 3 \pmod 8$. In order to obtain

the other odd numbers, we would require a further multiplier of 5 (mod 8). Greenberger used five multipliers which should scramble the sequence even further.

Hutchinson (1966) suggested a generator which revived Lehmer's original proposal. For a machine capable of representing an integer of maximum absolute value $2^p - 1$, the largest prime less than 2^p is used as the modulus. The value of a chosen is a primitive root of the prime. For the 7040, the modulus is $2^{35} - 31$, a is 5^5 or 5^{13}. For the 360, the modulus is $2^{31} - 1$, a is $2^7 + 1$. The distinct advantage of the Hutchinson method when compared to the more commonly used multiplicative-congruential method where $M = 2^p$ is that the least significant bits appear to be as random as the most significant bits. In order to understand the generator, let us look at the sequence generated by $a = 24$, $x_0 = 11$, modulus 31. We can assume that the word size for this example could accommodate a number as large as $2^5 - 1$. The sequence generated would be

11	16	12	9	30	7	13	2
17	5	27	28	21	8	6	20
15	19	22	1	24	18	29	14
26	4	3	10	23	25	11	

If we examine the differences between successive elements of the series, we notice that the differences are

5	27	28	21	8	6	...

This sequence is exactly the same sequence of numbers generated except that the starting value has changed. Similarly, if we examine the differences of lag 2, the same phenomenon occurs. In Table 3.4.3, the lag is specified as well as the first few terms of the sequence of differences.

Lag	Sequence			
1	5	27	28	21
2	1	24	18	29
3	29	14	26	4
4	19	22	1	24
5	27	28	21	8
6	2	17	5	27
7	22	1	24	18
8	6	20	15	19
9	25	11	16	12
10	16	12	9	30
11	17	5	27	28
12	10	23	25	11
13	28	21	8	6
14	26	4	3	10
15	9	30	7	13
16	4	3	10	23
17	8	6	20	15
18	11	16	12	9
19	21	8	6	20
20	13	2	17	5
21	7	13	2	17
22	18	29	14	26
23	3	10	23	25
24	15	19	22	1
25	24	18	29	14
26	23	25	11	16
27	30	7	13	2
28	12	9	30	7
29	14	26	4	3

Table 3.4.3 - First Four Members of Sequences with Lags up to 29

The significance of the results of Table 3.4.3 are not yet clear and might prove interesting for further study.

3.6 Concluding Remarks

It would be worth stating some pertinent remarks concerning pseudo-random number generators. Hull and Dobell (1964) point out that any generators which pass certain tests are not completely guaranteed of being acceptable. Some sequences of numbers may be perfectly acceptable as far as the tests are concerned, but will fail in the particular application for which they are being used. It is imperative that we be able to at least have a general idea of the expected answer before attempting the use of pseudo-random numbers. Lehmer's definition of a pseudo-random sequence as given by Taussky and Todd (1955) is: "A vague notion embodying the idea of a sequence in which each term is unpredictable to the initiated and whose digits pass a certain number of tests, traditional with statisticians and depending somewhat on the uses to which the sequence is to be put".

CHAPTER IV

THE GENERATION OF NON-UNIFORMLY DISTRIBUTED RANDOM VARIABLES

4.1 Introduction

The main purpose of this chapter is to examine methods for converting a uniformly distributed random variable to a normally distributed random variable. In addition, two other distributions of interest - the exponential distribution and the Poisson distribution will be considered.

First of all, however, let us consider the transformation of a given random variable ξ with a known probability density function to a random variable η with another probability density function. The following well known theorem from Freund (1962) is fundamental to the material discussed in this chapter.

Theorem 4.1.1: If the probability density function of $x \varepsilon \xi$ is given by $f(x)$ and the function given by $y = h(x)$ is differentiable and either increasing or decreasing for all values within the range of ξ, then the probability density function of $y \varepsilon \eta$ is given by

$$g(y) = f(x) \left|\frac{dx}{dy}\right| , \qquad (4.1.1)$$

where $\frac{dx}{dy} \neq 0$.

Proof: Let $F_\xi(a)$ represent the value of the distribution function of ξ at a . Then, the probability that ξ assumes a value less than or equal to a is

$$F_\xi(a) = P(\xi \leq a) = \int_{-\infty}^{a} f(x)dx \qquad (4.1.2)$$

We shall assume that $y = h(x)$ is increasing. (The proof may be modified when $h(x)$ is decreasing). Then, $F_\xi(a)$ also represents the probability that η assumes a value less than or equal to $h(a)$, i.e.,

$$F_\eta(h(a)) = \int_{-\infty}^{a} f(x)dx \ . \qquad (4.1.3)$$

If the change of variable $y = h(x)$, $x = H(y)$ is performed in (4.1.3), then

$$F_\eta(h(a)) = \int_{-\infty}^{h(a)} f(H(y))\ H'(y)dy \qquad (4.1.4)$$

for any real number $h(a)$ within the range of η. Therefore, if $H'(y)$ exists, the integrand in equation (4.1.4) gives the probability density function of η. That is, the probability density function of η is given by

$$g(y) = f(H(y))\, H'(y) ,$$

$$\frac{1}{H'(y)} \neq 0 \qquad (4.1.5)$$

which can be written

$$g(y) = f(x) \frac{dx}{dy}$$

since $\frac{1}{H'(y)} = 0$, $H(y) = x$ and $H'(y) = \frac{dx}{dy}$.

Keeping (1962) shows that if $F(x)$ is the distribution function of X, and if the transformation $Y = F(X)$ is made, then Y has a uniform distribution on (0,1). In 4.3, we shall see how the preceding theory can be used

to derive the exponential distribution.

4.2 The Normal Distribution

Since normally distributed random variables are important in statistics, conversions from a uniformly distributed random variable to a normally distributed random variable have been studied and reviewed extensively in the literature. Among the most important conversion techniques are those considered by Box and Muller (1958), Hastings (1955), Juncosa (1953), Marsaglia (1964), and von Neumann (1951).

Von Neumann's technique depends on an acceptance-rejection procedure. In order to generate normal deviates X in the region $-b \leq X \leq b$, two uniform random deviates U_1 and U_2 are generated. Then $Y = -2b^2(U_1-.5)^2$ is computed. If $\log_e U_2 \leq Y$, then $X = b(2U_1-1)$ is used as a normal deviate. Otherwise, if $\log_e U_2 > Y$, then the pair (U_1,U_2) is rejected and another pair of uniform random deviates would be generated and the procedure described above is repeated.

The von Neumann approach is inefficient for generating normal deviates especially when values beyond three standard deviations from the mean are needed. The inefficiency is recognized as soon as an expression is found for the probability that a pair (U_1,U_2) will be

used to generate a normal deviate. The probability may be expressed as

$$P\{U_2 \leq e^{-2b^2(U_1-.5)^2}\} = \int_0^1 e^{-2b^2(U_1-.5)^2}\, dU_1 \qquad (4.2.1)$$

which asymptotically approaches $(1/b)\sqrt{\pi/2}$. If b becomes small, then the probability increases proportionately.

The Hasting's approach uses a Padé approximation to transform a uniform random deviate to a normal random deviate. The most accurate method proposed obtains a normal random deviate X from a uniform random deviate q using

$$X = X^*(q) = \eta - \left\{\frac{a_0+a_1\eta+a_2\eta^2}{1+b_0+b_1\eta+b_2\eta^2}\right\} \qquad (4.2.2)$$

where $a_0 = 2.515517$ $\quad b_0 = 1.432788$

$a_1 = 0.802853$ $\quad b_1 = 0.189269$

$a_2 = 0.010328$ $\quad b_2 = 0.001308$

$$\eta = \sqrt{\log_e (1/q)^2}$$

and

$$q = \frac{1}{\sqrt{2\pi}} \int_{x(q)}^{\infty} e^{-\frac{1}{2}t^2} \, dt$$

$$0 \leq q \leq .5 \quad .$$

If q lies in the interval, $.5 < q \leq 1$, then a transformation must be performed on q to reduce it to the $[0,5]$. The transformation used is

$$q = 1 - q \quad .$$

The value of X is found from (4.2.2) and the normal random deviate used is the value of (4.2.2) with the sign changed.

Hasting's approach gives results which are accurate to within $\pm 4 \times 10^{-4}$ of the correct result. However, the time taken to generate normal deviates is greater than for some other methods. In order to compensate for

this, Hasting's derived another Padé approximation using fewer constants. The conversion is given by

$$X^*(q) = \eta - \{\frac{a_0+a_1\eta}{1+b_0\eta+b_1\eta^2}\} \qquad (4.2.3)$$

where $a_0 = 2.30753$ $b_0 = 0.99229$

$a_1 = 0.27061$ $b_1 = 0.04481$

and q and η are given by equation (4.2.2). The decrease in time is compensated by the difference in the maximum absolute error. In (4.2.3), the maximum absolute error is 3×10^{-3}.

Juncosa's approach makes use of the Central Limit Theorem which may be stated as follows (Keeping (1962)).

Theorem 4.2.1: Let $X_1, X_2, X_3, \ldots, X_\eta$ be independent random variates all having the same distribution with mean μ and variance σ^2, but not necessarily normal. Let the standarized variate corresponding to X_j be

$$Z_j = \frac{X_j - \mu}{\sigma} \quad , \qquad (4.2.4)$$

and let Y_n be defined by

$$Y_n = \frac{\sum_{j=1}^{n} Z_j}{\sqrt{n}} = \sqrt{n}\ \bar{Z} \quad , \qquad (4.2.5)$$

where $\bar{Z}$ is the arithmetic mean of the Z_j . Then, as $n \to \infty$, Y_n tends to a standard normal variate.

For the uniform distribution on (0,1), it can be shown that

$$\mu = 1/2$$

and

$$\sigma^2 = 1/12$$

Therefore, from (4.2.4),

$$Z_j = (X_j - .5)\ \sqrt{12} \qquad (4.2.6)$$

and

$$Y_n = \frac{\sum_{j=1}^{n} (X_j - .5)\sqrt{12}}{\sqrt{n}} \qquad (4.2.7)$$

The Juncosa approach is reasonably fast and requires little memory space. However, for every random normal deviate generated, a sufficiently large number of random uniform deviates must be available. In Juncosa's report, sixty-four uniform random numbers were generated for each normal variate. The Appendix shows plots for some values of n.

Marsaglia's method is particularly fast. The normal distribution is expressed as a linear combination of three separate functions,

$$g(x) = 0.9578g_1(x) + 0.0395g_2(x) + 0.0027g_3(x) \qquad (4.2.8)$$

where

$$g(x) = \frac{2e^{-\frac{1}{2}x^2}}{\sqrt{2\pi}}$$

The distributions of $g(x)$, $g_1(x)$, $g_2(x)$ and $g_3(x)$ are shown in Figures 4.2.4, 4.2.1, 4.2.2, and 4.2.3 respectively. (See Marsaglia (1964)).

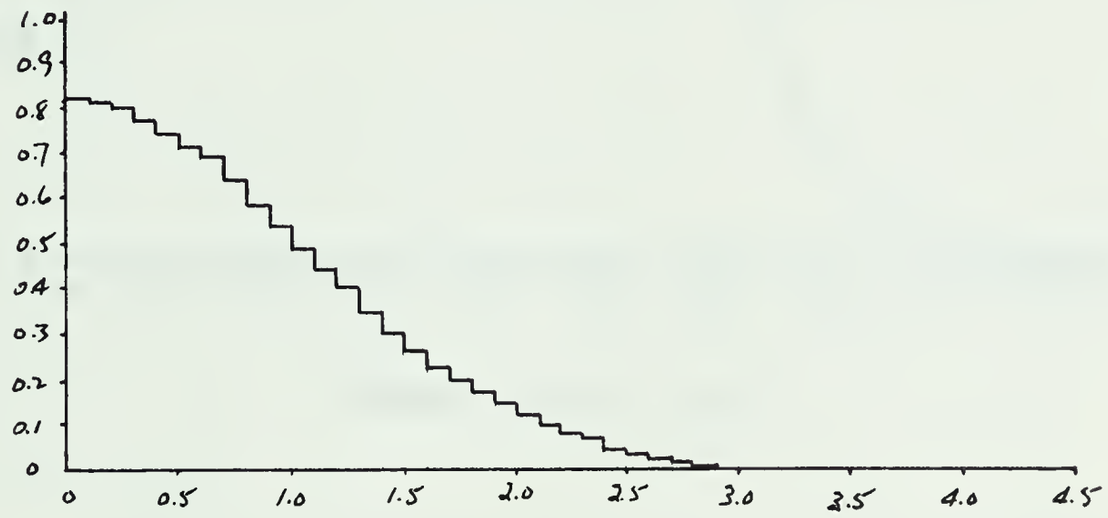


Figure 4.2.1 - $g_1(x)$

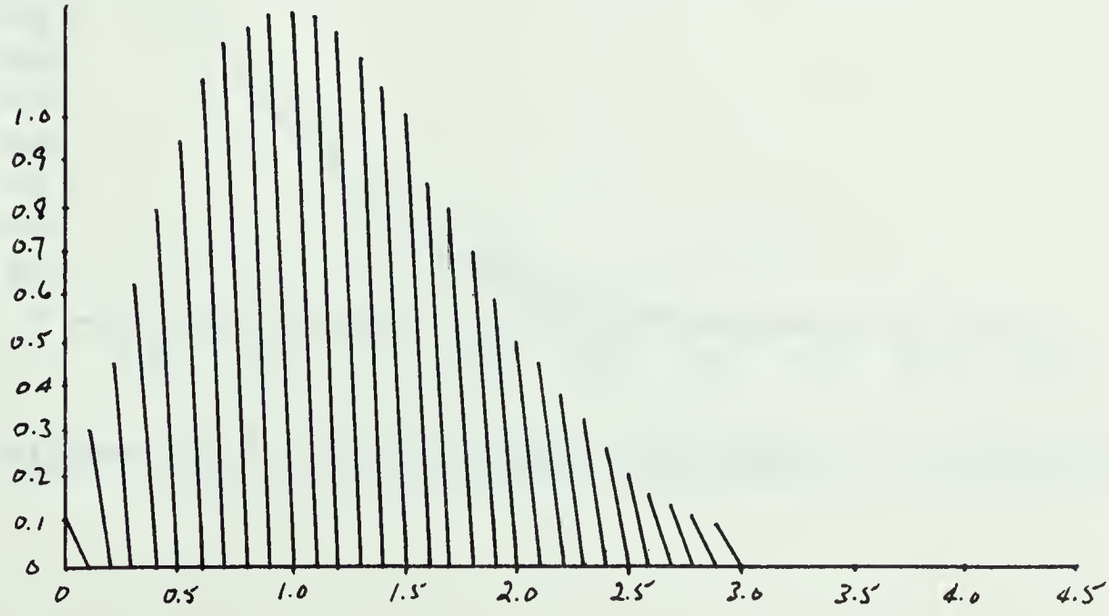


Figure 4.2.2 - $g_2(x)$

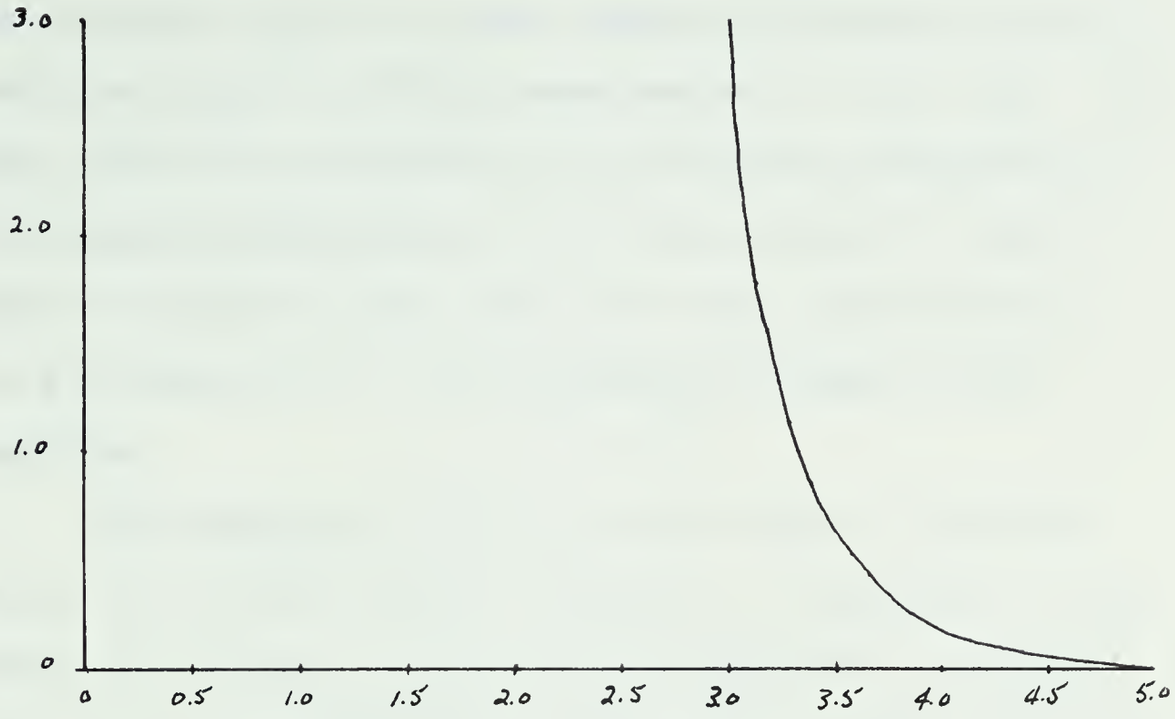


Figure 4.2.3 - $g_3(x)$

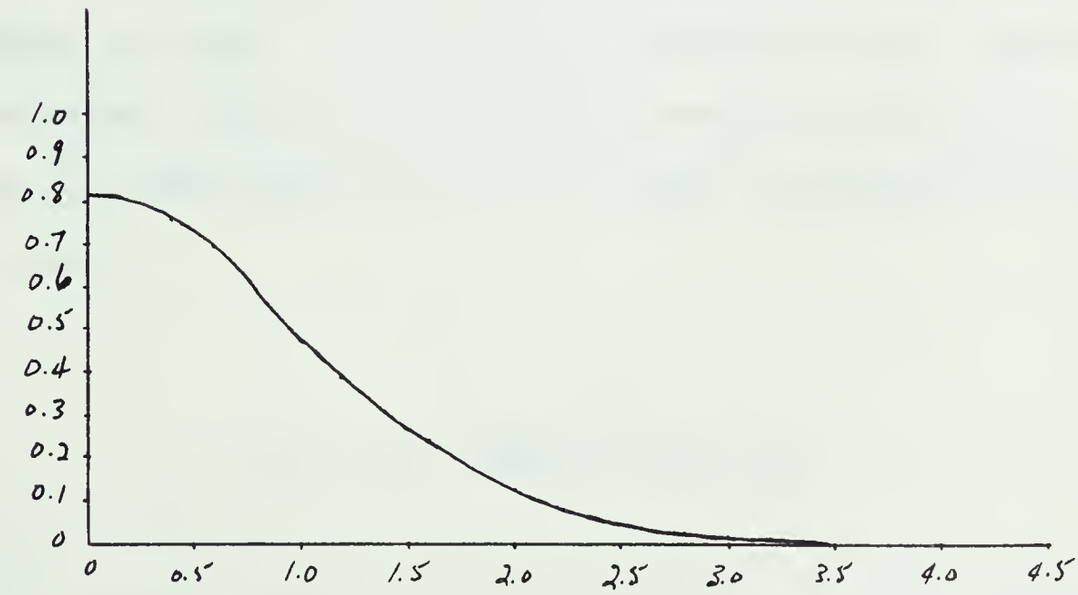


Figure 4.2.4 - $g(x)=.9578g_1(x)+.0395g_2(x)+.0027g_3(x)$

The function $g_1(x)$ is very simple to evaluate and is used ninety-six to ninety-seven percent of the time. The other three or four percent of the time a modification involving a complicated set of instructions is used to make the function agree with the normal distribution. The flow chart for a binary machine is shown in the Appendix.

The Marsaglia method is fast since it consists mainly of a table look-up. However, it requires a large amount of storage to store all the constants necessary. The method could be made to generate the entire range of values if a plus or a minus sign were placed randomly in the result

The Box and Muller approach converts two uniform random deviates to two normal random deviates directly. The method takes the two uniform random deviates U_1 and U_2 from the interval (0,1) and transforms them as follows:

$$X_1 = (-2 \log_e U_1)^{\frac{1}{2}} \cos 2\pi U_2$$

$$X_2 = (-2 \log_e U_1)^{\frac{1}{2}} \sin 2\pi U_2 \qquad (4.2.9)$$

(X_1,X_2) will then be a pair of independent normal random variables with mean 0 and variance 1.

In order to justify the above, from (4.2.9), the inverse relationships are

$$U_1 = e^{\frac{-X_1^2 - X_2^2}{2}}$$

$$U_2 = -\frac{1}{2\pi} \arctan\left(\frac{X_2}{X_1}\right) \qquad (4.2.10)$$

It can be shown that the joint density of X_1, X_2 is

$$f(X_1,X_2) = \frac{1}{2\pi} e^{-\frac{X_1^2 + X_2^2}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{X_1^2}{2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{X_2^2}{2}}$$

$$= f(X_1)f(X_2) \ . \qquad (4.2.11)$$

which gives (X_1,X_2) as two independent normally distributed random deviates with mean 0 and variance 1.

The direct approach of Box and Muller gives better accuracy than comparable methods and involves little memory space. An added advantage to this method is that values beyond three standard deviations of the mean are reliable. Furthermore, the transformation from uniform to normal is exact and every uniform variable can be transformed.

4.3 The Exponential Distribution

From 4.1, a method can be derived to convert a uniform random variable ξ to an exponentially distributed random variable η.

Let $f(x)$ be the probability density function of the uniformly distributed random variable ξ, and let $h(y)$ be the probability density function of the exponentially distributed random variable η. Then

$$h(y) = \lambda e^{-\lambda y} \quad . \qquad (4.3.1)$$

Let ξ_i represent a uniformly distributed pseudo-random number from which we wish to generate η_i, an exponentially distributed random number. From 4.1.2 and 4.1.4

$$\xi_i = F_\eta(\eta_i) = \int_{-\infty}^{\eta_i} \lambda e^{-\lambda y}\, dy \quad , \qquad (4.3.2)$$

$$\xi_i = -e^{-\lambda \eta_i} + 1 \quad .$$

Therefore

$$1 - \xi_i = e^{-\lambda \eta_i}$$

and

$$\eta_i = \frac{1}{\lambda} \log_e (1-\xi_i) \qquad (4.3.3)$$

If the numbers ξ_i are uniformly distributed on (0,1), then, by solving (4.3.3), it is possible to compute a sequence of random variables η_i with the exponential distribution.

4.4 Poisson Distribution

We will now turn our attention to the generation of a discrete random variable distribution by considering the Poisson distribution. The probability distribution for Poisson is

$$f(x,\lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \qquad (4.4.1)$$

where

$$x = 0,1,2,\ldots$$

and

$$\lambda = n\theta \ , \ (n\to\infty, \theta\to 0) \ , \text{ is constant} \ .$$

The cumulative probability density distribution is given by

$$F(x,\lambda) = \sum_{i=0}^{x} \frac{\lambda^i e^{-\lambda}}{i!} \qquad (4.4.2)$$

In order to generate random variables satisfying a Poisson distribution from a continuous uniform distribution, we would proceed in the following manner. We would generate a uniform deviate R_i in (0,1). If

$$T_{j-1} \leq R_i < T_j$$

where

$$T_k = \sum_{i=0}^{k} \frac{\lambda^i e^{-\lambda}}{i!} ,$$

then the value j is used as the Poisson deviate. This procedure would be repeated for a sufficient number of trials. The distribution of the j's would then satisfy the Poisson distribution for the particular λ chosen.

CHAPTER V

SOME SIMPLE APPLICATIONS

5.1 Buffon's Needle

A rough approximation to the value π may be determined by considering Buffon's needle problem (see Keeping (1962)). The problem consists of a needle of length ℓ thrown onto a table which has been ruled with equidistant parallel lines a distance a ($\ell<a$) apart. Buffon, the celebrated French naturalist posed the problem: What is the probability that the needle will intersect one of the lines? We may form an empirical formula for the probability as follows. If we take the x-axis as being one of the parallel lines, then the x-coordinate of the centre of the needle will be immaterial, i.e., we need only consider the y-coordinate of the centre of the needle and the angle θ which the needle makes with the x-axis. The needle will cross a parallel straight line if the distance y from the centre of the needle to the nearest line is less than or equal to $\frac{1}{2}\ell \sin\theta$. (See Figure 5.1.1).

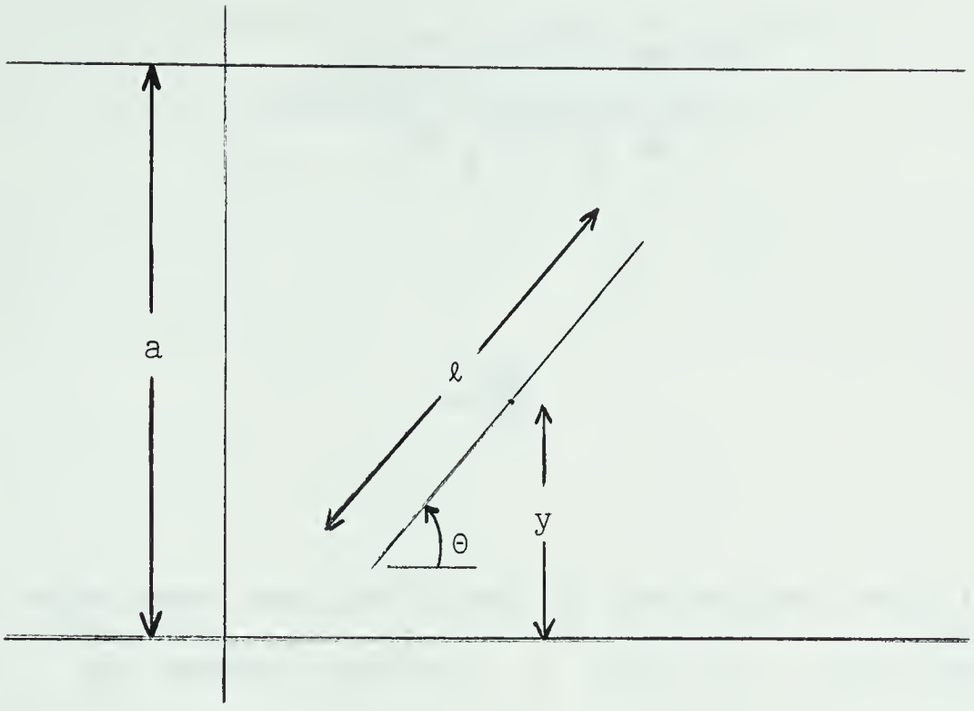


Figure 5.1.1 - Buffon's Needle Problem

The probability that the needle will cross the nearest straight line is given by

$$p = \frac{\int_0^{\pi} \int_0^{\frac{1}{2}\ell \sin\Theta} f(\Theta,y)\, dy\, d\Theta}{\int_0^{\pi} \int_0^{a/2} f(\Theta,y)\, dy\, d\Theta} \qquad (5.1.1)$$

Since we have assumed that the needle has been tossed at random, then all values of Θ and y are equally likely and, as a result, $f(\Theta,y)$ is constant and (5.1.1) becomes

$$p = \frac{\int_0^{\pi} \int_0^{\frac{1}{2}\ell \sin\theta} dy\, d\theta}{\int_0^{\pi} \int_0^{a/2} dy\, d\theta} \qquad (5.1.2)$$

$$= \frac{2\ell}{\pi a} \qquad (5.1.3)$$

This experiment was performed in the manner described below. Two random numbers, y , and y_2 , were generated within the interval (0,1). The centre of the needle was assigned to coordinate y_1 . In order to calculate the value of θ , y_2 was converted to radians by setting $\theta = \pi y_2$. The test to compare y_1 to $\frac{1}{2}\ell \sin\theta$ was performed. If $y_1 > \frac{1}{2}\ell \sin\theta$, then the total number of throws was incremented by one. If $y_1 \leq \frac{1}{2}\ell \sin\theta$, then the total number of throws was incremented by one and the total number of times a line was crossed was incremented by one. When a sufficiently large number of throws was performed, the ratio, p , of crosses to the total tosses was computed. The value of π was approximated using

$$\pi \simeq \frac{2\ell}{pa} \qquad (5.1.4)$$

Table 5.1.1 shows the results from one computer run with the generator $x_{i+1} \equiv 5^{13}x_i \pmod{2^{35}}$.

Number of Tosses	Estimate for π		
	ℓ = .011	ℓ = .005	ℓ = .09
	a = .02	a = .01	a = .10
10000	3.1321	3.1939	3.1518
20000	3.1464	3.1615	3.1422
30000	3.1384	3.1695	3.1522
40000	3.1208	3.1638	3.1564
50000	3.1219	3.1643	3.1565
60000	3.1260	3.1585	3.1571
70000	3.1191	3.1598	3.1517
80000	3.1161	3.1535	3.1535
90000	3.1185	3.1502	3.1515
100000	3.1215	3.1542	3.1521

Table 5.1.1 - Estimates for π Using the Buffon Needle Technique

From Table 5.1.1, it can be seen that only a very rough approximation to π may be found using the Buffon approach. If it were possible to generate infinitely many random numbers, then we could, theoretically, calculate π to as many significant digits as we desire. However, as pointed out in Hull and Dobell (1962), the solution is approached

with an error of order $n^{\frac{1}{2}}$, where n is the number of trials. This means that an additional significant digit requires n^2 as many computations as the previous digit. It can be seen from Table 5.1.1 that only two significant digits result from 100,000 tosses of the needle.

5.2 Chuck-a-Luck

One example of a game of chance is the familiar chuck-a-luck. The game is played with three ordinary dice which are tossed. A player is paid according to the number of dice which show his bet. Table 5.2.1 shows the probabilities and the return to a player who has bet one dollar.

Result of Throw	Probability	Payoff
No 6's	.579	-$1
One 6	.347	$1
Two 6's	.069	$2
Three 6's	.005	$3
	1.000	-$. 0787

Table 5.2.1 - Payoff to Player Betting on 6

From Table 5.2.1, it can be seen that a player will expect to lose 7.87 cents whenever the dice are tossed. From this, we can show that a player who starts with x dollars can expect to have lost all his money after

$$N = \frac{x}{.0787} \qquad (5.2.1)$$

throws of the three dice. The expected duration of the game is shown in Table 5.2.2.

Initial Amount	Expected Duration	Mimimum Observed	Maximum Observed
$5	63.5	5	385
$10	127.05	12	465
$20	254.1	42	2057

Table 5.2.2 - Expected Duration of Chuck-a-Luck

In order to simulate chuck-a-luck, four random numbers in the interval (0,1) were generated. The interval was divided into six subintervals to determine the player's bet and the result of the throw of the three dice. The player's payoff was determined according to the number of dice showing his bet. The results for a simulation of fifty games are shown in the Appendix.

5.3 Integration Under a Curve

The third example of Monte Carlo techniques is integration under a curve, i.e., $\int_a^b f(x)dx$. The method may be developed in the following manner.

The maximum value of the function $f(x)$ within the interval is computed and given a value Max . The value of $b-a$ is computed (where a and b are the lower and upper limits of integration respectively) and given a value of T . Two random coordinates (x,y) are generated within the interval $(0,1)$ and transformed as follows:

$$x \leftarrow x \text{ Max}$$

$$y \leftarrow y \ T \ .$$

If $y \leq f(x)$, then the total number of points on or below the curve is incremented by one. Otherwise, the total number of points above the curve is incremented by one. The ratio of the value

$$R = \frac{\text{Total points on or below the curve}}{\text{Total points below, on, or above the curve}}$$

is computed. The area may be approximated by

$$\text{Area} \simeq R \ T \ \text{Max} \ .$$

The number of points needed to calculate the next significant digit is n^2 as many as for the previous digit - a situation analogous to that of Buffon's needle.

5.4 Hypothesis Testing

Specifically, we are concerned with the probability of analyzing a disease D correctly when given a set of symptoms $S = \{s_1, s_2, \ldots, s_k\}$. By Bayes' Theorem (see Freund (1962)), we know that

$$P(D|S) = \frac{P(D)}{P(S)} P(S|D) . \qquad (5.4.1)$$

Two different cases could be examined - either the symptoms are independent of each other, or that some of the symptoms are pairwise dependent. For simplicity, we will consider three symptoms s_1, s_2, s_3 and associated estimates of the conditional probabilities p_1, p_2, p_3 of the disease D given the symptoms s_1, s_2, s_3, e.g., P_1 is an estimate of $p_1 = P(s_1|D)$ and P_{23} is an estimate of $p_{23} = P(s_2 \cap s_3|D)$.

In the simulation, let us consider the second case, i.e., two symptoms 2 and 3 are pairwise dependent. Then, if we assume

$$P_1 = \frac{f_1}{N} \simeq .5$$

where f_1 is the frequency of the symptom s_1 and N is the number of observations and

$$P_{23} = \frac{f_{2\cap 3}}{N} \quad .$$

With different values of p_{23} for different samples, we should be able to obtain the distributions of P_1 and P_{23}. We shall assume that p_1 is binomial with mean P_1 and variance

$$\frac{P_1(1-P_1)}{N} \quad ,$$

and p_{23} is binomial with mean P_{23} and variance

$$\frac{P_{23}(1-P_{23})}{N} \quad .$$

In addition to obtaining the distributions, we would like to examine the results of two different probability estimates

$$p' = \sqrt{p_{12}p_{13}p_{23}} \qquad (5.4.1)$$

and

$$p'' = \frac{1}{3}\,(p_1p_{23}+p_2p_{13}+p_3p_{12}) \qquad (5.4.2)$$

of detecting the disease D given the symptoms s_1, s_2, s_3.

The simulation was run on the 7040 computer for six different conditions. The values p_1 and p_2 were set constant at .5, however, $p_{3|2}$ was varied as .5, .7, and .9. The sample frequencies of P' and P'' were calculated using five hundred simulations for sample sizes of twenty or fifty observations. The simulation consisted of generating pseudo-random numbers with a uniform distribution, determining whether a symptom s_1, s_2, or s_3 was present and tallying the frequencies. The results are shown in Table 5.4.1.

Sample Size	$P_{3\|2}$	$\bar{P}'$	$\bar{P}''$
20	.5	.1298	.1295
20	.7	.1462	.1409
20	.9	.1664	.1567
50	.5	.1268	.1266
50	.7	.1502	.1443
50	.9	.1662	.1569

Table 5.4.1 - Results from Simulation of Disease Detection for Symptoms s_1, s_2, s_3 .

The simulation may be carried further to examine binomial product distributions of P' and P'' for the entire set of 500 observations.

It might be of interest to examine the bias in the two estimates (5.4.1) and (5.4.2) of P_{123} . If we assume that 1 and 2 are independent, i.e., $P_{12} = P_1P_2$, but that, in general, 2 and 3, are not, i.e., $P_{23} = P_2P_{3|2}$. Then, p' and p'' are both estimates of $P_{123} = P_1P_2P_{3|2}$.

Substituting population values in (5.4.1), we get

$$\sqrt{P_{12}P_{23}P_{13}} = \sqrt{(P_1P_2)(P_2P_{3|2})(P_1P_3)}$$

$$= \sqrt{(P_1P_2P_{3|2})^2(P_3/P_{3|2})}$$

$$= P_{123}\sqrt{P_3/P_{3|2}}$$

The expression under the radical is the bias which is 1 if and only if 2 and 3 are independent, i.e., if and only if $P_3 = P_{3|2}$.

Substituting population values in (5.4.2), we get

$$\tfrac{1}{3}(P_1P_{23}+P_2P_{13}+P_3P_{12}) = \tfrac{1}{3}(P_1P_2P_{3|2}+P_2P_1P_3+P_3P_1P_2)$$

$$= \tfrac{1}{3}P_1P_2(P_{3|2}+2P_3)$$

$$= P_{123}\left(\frac{P_{3|2}+2P_3}{3P_{3|2}}\right).$$

Again, the estimate is unbiased if and only if $P_3 = P_{3|2}$.

Both bias terms can be related to the parameters of the simulation by substituting

$$P_3 = P_2 P_{3|2} + (1-P_2)(1-P_{3|2}) .$$

BIBLIOGRAPHY

Allard, J.L., A.R. Dobell and T.E. Hull, 1963. "Mixed Congruential Random Number Generators for Decimal Machines", J. Assoc. Comp. Mach., 10:131-141.

Barnett, V.D., 1962. "The Behaviour of Pseudo-Random Sequences Generated on Computers by the Multiplicative Congruential Method", Math. Comp., 16:63-69

Bofinger, E. and V.J. Bofinger, 1958. "On a Periodic Property of Pseudo-Random Sequences", J. Assoc. Comp. Mach., 5:261-265.

Box, G.E.P. and M.E. Muller, 1958. "A Note on the Generation of Random Normal Deviates", Annals Math. Stat., 29:610-611.

Certaine, J., 1958. "On Sequences of Pseudo-Random Numbers of Maximal Length", J. Assoc. Comp. Mach., 5:353-356.

Chambers, R.P., 1967. "Random-Number Generation on Digital Computers", IEEE Spectrum, 4:48-56.

Coveyou, R.R., 1960. "Serial Correlation in the Generation of Pseudo-Random Numbers", J. Assoc. Comp. Mach., 7:72-74.

Fisher, R.A. and F. Yates, 1938. "Statistical Tables for Biological, Agricultural and Medical Research", Oliver and Boyd, London.

Fisser, H., 1961. "Some Tests Applied to Pseudo-Random Numbers Generated by V. Horner's Rule", Numer. Math., 3:247-249.

Franel, J., 1917. "Vierteljahrschrift der Naturforschenden Gessalschaft in Zurich", 62:268.

Freund, J.E., 1962. "Mathematical Statistics", Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Good, I.J., 1953. "The Serial Test for Sampling Numbers and Other Tests for Randomness", Proc. Camb. Phil. Soc., 49:276-284.

Good, I.J., 1957. "On the Serial Test for Random Sequences", Annals Math. Stat., 28:109-110.

Gorenstein, S., 1967. "Testing a Random Number Generator", Comm. Assoc. Comp. Mach., 10:111-118.

Green, B.F. Jr., J.E. Smith and L. Klem, 1959. "Empirical Tests of an Additive Random Number Generator", J. Assoc. Comp. Mach., 6:527-537.

Greenberger, M., 1961. "An A Priori Determination of Serial Correlation in Computer Generated Random Numbers", Math. Comp., 15:383-389.

Greenberger, M., 1961. "Notes on a New Pseudo-Random Number Generator", J. Assoc. Comp. Mach., 8:163-167.

Greenberger, M., 1965. "Method in Randomness", Comm. Assoc. Comp. Mach., 8:177-179.

Gruenberger, F. and A.M. Mark, 1951. "The d^2 Test of Random Digits", Math. Tables Aid Comp., 5:109-110.

Hammersley, J.M. and D.C. Handscomb, 1964. "Monte Carlo Methods", John Wiley and Sons, Inc., New York.

Hamming, R.W., 1962. "Numerical Methods for Scientists and Engineers", McGraw-Hill Book Company, Inc., New York.

Hastings, C. Jr., 1955. "Approximation for Digital Computers", Princeton University Press, Princeton, New Jersey.

Hull, T.E. and A.R. Dobell, 1962. "Random Number Generators", SIAM Review, 4:230-254.

Hull, T.E. and A.R. Dobell, 1964. "Mixed Congruential Random Number Generators for Binary Machines", J. Assoc. Comp. Mach., 11:31-40.

Hutchinson, D.W., 1966. "A New Pseudorandom Number Generator", Comm. Assoc. Comp. Mach., 9:432-433.

International Business Machines Corporation, 1959. "Random Number Generation and Testing", Reference Manual C20-8011, New York.

Juncosa, M.L., 1953. "Random Number Generation on the BRL High-Speed Computing Machines", Ballistic Research Laboratories, Report No. 855, Aberdeen Proving Ground

Keeping, E.S., 1962. "Introduction to Statistical Inference", D. Van Norstrand Company, Inc., Princeton, New Jersey.

Kendall, M.G. and B. Babington-Smith, 1938. "Randomness and Random Sampling Numbers", J. Royal Stat. Soc., 101:147-166.

Kendall, M.G. and B. Babington-Smith, 1939. "Second Paper on Random Sampling Numbers", J. Royal Stat. Soc., 6:51-61.

Kendall, M.G. and B. Babington-Smith, 1939. "Tables of Random Sampling Numbers", Tracts for Computers, 24.

McCracken, D.D., 1955. "The Monte Carlo Method", Scientific American, 192:90-96.

MacDonell, W.R., 1901. "On Criminal Anthropometry and the Identification of Criminals", Biometrika, 1:219.

Marsaglia, G., M.O. MacLaren and T.A. Bray, 1964. "A Fast Procedure for Generating Normal Random Deviates", Comm. Assoc. Comp. Mach., 7:4-10.

Meyer, H.A., (editor), 1956. "Symposium on Monte Carlo Methods", John Wiley and Sons, Inc., New York.

Muller, M.E., 1959. "A Comparison of Methods for Generating Normal Deviates on Digital Computers", J. Assoc. Comp. Mach., 6:376-383.

Ore, O., 1948. "Number Theory and Its History", McGraw-Hill Book Company, Inc., New York.

Pawlak, Z., 1956. "Flip-Flop as a Generator of Random Binary Digits", Math. Tables Aid Comp.,10:28-30.

Peach, P., 1961. "Bias in Pseudo-Random Numbers", J. Am. Stat. Assoc., 56:610-618.

Ralston, A. and H.S. Wilf, (editors), 1960. "Mathematical Methods for Digital Computers", John Wiley and Sons, Inc., New York.

RAND Corporation, 1955. "A Million Random Digits with 100,000 Normal Deviates", The Free Press, New York.

Richtmeyer, R.D., 1961. "Monte Carlo Methods", Proc. Symposium Appl. Math., 11:190-205.

Rotenberg, A., 1960. "A New Pseudo-Random Number Generator", J. Assoc. Comp. Mach., 7:75-77.

Shreider, Y.A., 1964. "Method of Statistical Testing", Elsevier Publishing Company, New York.

Shreider, Y.A., 1966. "The Monte Carlo Method", Pergamon Press, New York.

Stockmal, F., 1964. "Calculations with Pseudo-Random Numbers", J. Assoc. Comp. Mach., 11:41-52.

Student, 1908. "The Probable Error of a Mean", Biometrika, 6:1-25.

Taussky, O. and J. Todd, 1956. "Generation and Testing of Pseudo-Random Numbers", Symposium on Monte Carlo Methods, H.A. Meyer, (editor), John Wiley and Sons, Inc., New York.

Thompson, A.J., 1927. "Logarithmetica Britannica", Cambridge University Press.

Thomson, W.E., 1959. "ERNIE - A Mathematical and Statistical Analysis", J. Royal Stat. Soc., A122:301-324.

Tippett, L.H.C., 1925. "On the Extreme Individuals and the Range of Samples Taken from a Normal Population", Biometrika, 17:364-397.

Tippett, L.H.C., 1927. "Random Sampling Numbers", Tracts for Computers, 15.

Tocher, K.D., 1954. "The Application of Automatic Computers to Sampling Experiments", J. Royal Stat. Soc., 16:39-61.

Tocher, K.D., 1963. "The Art of Simulation", English Universities Press, London.

Todd, J., (editor), 1962. "Survey of Numerical Analysis", McGraw-Hill Book Company, Inc., New York.

U.S. National Bureau of Standards, 1951. "Monte Carlo Method", AMS 12, Washington.

Von Neumann, J., 1951. "Various Techniques Used in Connection with Random Digits", Monte Carlo Method, Nat. Bur. Stand., AMS 12:36-38.

APPENDIX

Two multiplicative-congruential pseudo-random number generators are examined in the Appendix. The first uses a modulus of 2^{35}, the second a modulus of $2^{35}-31$ (the largest prime less than 2^{35}). The MAP source statements for both generators are shown. Either generator may be called from a FORTRAN mainline program in the same manner as a FORTRAN function subprogram, i.e., Y = RANDOM (X), X is the initial value to be used. The pseudo-random number is returned as a real constant in the interval (0,1). If integer pseudo-random numbers are required between 1 and M ($M=2^{35}$ or $2^{35}-31$), the revisions necessary are shown.

The frequencies and serial frequencies for blocks of 10000 random numbers are shown in the next section of the Appendix. It was seen that both methods gave short cycles in the least significant digits, so only nine or ten octal digits were examined. In all cases, the bit for the sign was ignored for the reasons stated in section 2.5. Both generators, with the appropriate multipliers, gave acceptable chi-square values.

The results from Gorenstein's test and the d^2 test are tabulated in the next section. In either test, the multiplicative-congruential method gave acceptable chi-square values.

Plots of the Box and Muller normal transformation

and Juncosa's transformation for sixteen and sixty-four elements are given.

The last section of the Appendix displays the results of the applications discussed in Chapter V.

```
$IBMAP MULCON
*
*       MULTIPLICATIVE-CONGRUENTIAL PSEUDO RANDOM NUMBER GENERATOR
*
*       ... CALLING SEQUENCE FROM FORTRAN MAINLINE IS ...
*                       Y=RANCON(X)
*                               WHERE X IS THE STARTING VALUE
*                               CHOSEN BY THE PROGRAMMER.
*                               IF X IS EVEN, IT IS MADE ODD.
*
*
*               THE PSEUDO-RANDOM NUMBER RETURNED HAS A VALUE IN
*                               THE OPEN INTERVAL (0,1).
*
        ENTRY   RANCON          DEFINE THE ENTRY POINT
RANCON  SAVE    1,2,4           SAVE THE RETURN LINKAGE
        CLA     SWITCH          TEST FOR FIRST TIME
        TPL     NOTFST            THROUGH
*
*       FIRST TIME THROUGH GENERATOR - INITIALIZE X
*
        MSP     SWITCH          RESET SWITCH
        CLA*    3,4             GET THE INITIAL VALUE OF X
        CRA     =0CCCCCCCCCCC01 MAKE X ODD
        STO     X               STORE INITIAL X
*
*       END OF INITIALIZATION - MAIN ROUTINE FOLLOWS
*
NOTFST  LDQ     X               PUT X INTO MQ
        MPY     =1220703125     MULTIPLY BY 5**13
        STQ     X               REPLACE OLD X WITH NEW X (MOD 2**35)
        CLA     X               PUT X INTO AC
        ARS     8              *SET UP X FOR FLOATING POINT
        CRA     FLOT           *INTRODUCE EXPONENT
        FAD     FLOT           *NORMALIZE MANTISSA
        RETURN  RANCON          RETURN TO MAINLINE WITH X IN AC
*
*       CONSTANTS
*
X       BSS     1               STORAGE FOR X
FLOT    OCT     2CCCCCCCCCCC    FLOATING POINT EXPONENT / NORMALIZER
SWITCH  OCT     4CCCCCCCCCCC0   INITIALIZE AS A NEGATIVE NUMBER
*
*       INSTRUCTION DESCRIPTIONS BEGINNING WITH * MAY BE DELETED
*               IF INTEGER VALUES BETWEEN 1 AND 2**35 ARE REQUIRED
*               INSTEAD OF FLOATING POINT NUMBERS
*
        END                     END OF SUBROUTINE
```

MAP Listing for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}}$

```
$IBMAP LEHMER
*
*       MULTIPLICATIVE-CONGRUENTIAL PSEUDO-RANDOM NUMBER GENERATOR
*               USING A MODULUS OF THE LARGEST PRIME POSSIBLE
*                       IN THE 7040
*
*       ... CALLING SEQUENCE FROM FORTRAN MAINLINE IS ...
*                       Y=RANLEH(X)
*
*               VALUE RETURNED IS SAME AS FOR MODULUS 2**35 PROGRAM
*
        ENTRY   RANLEH          DEFINE THE ENTRY POINT
RANLEH  SAVE    1,2,4           SAVE THE RETURN LINKAGE
        CLA     SWITCH          IS THIS THE FIRST TIME
        TPL     NOTFST          NO, TRANSFER TO NOTFST
*
*       INITIALIZATION FOR X
*
        MSP     SWITCH          RESET SWITCH
        CLA*    3,4             GET THE INITIAL X
        STO     X                AND STORE IT
*
*       ROUTINE TO CALCULATE NEW X
*
NOTFST  LDQ     X               PUT X INTO MQ
        MPY     =1220703125     MULTIPLY BY 5**13
        DVP     =037777777741 DIVIDE TO GET  MOD 2**35-31
        STO     X               STORE NEW X
        ARS     8              *SHIFT X TO FLOAT IT
        ORA     FLOT           *INSERT EXPONENT
        FAD     FLOT           *NORMALIZE IT
        FAD     FLOT           *NORMALIZE THE PSEUDO-RANDOM NUMBER
        RETURN  RANLEH          RETURN TO MAINLINE
*
*       CONSTANTS
*
X       BSS     1               STORAGE FOR X
FLOT    OCT     200000000000    FLOATER AND NORMALIZER
SWITCH  OCT     400000000000    SWITCH FOR FIRST TIME THROUGH
*
*       INSTRUCTIONS WITH * IN COL 30 MAY BE OMITTED IF AN INTEGER
*               BETWEEN 1 AND 2**35-31 IS DESIRED
*
        END                     END OF SUBROUTINE
```

MAP Listing for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}-31}$

SERIAL FREQUENCY FOR OCTAL POSITION 1

	0	1	2	3	4	5	6	7
C	148	149	164	153	147	179	153	165
1	152	138	156	155	161	176	159	141
2	149	168	172	147	152	167	159	147
3	180	170	150	170	129	130	189	172
4	164	141	150	157	147	144	145	143
5	149	161	147	170	156	150	164	163
6	167	158	160	170	161	168	153	142
7	149	153	162	168	138	146	157	150
	1258	1238	1261	1290	1191	1260	1279	1223

CHI-SQUARE FOR SERIAL FREQUENCY IS 58.52
CHI-SQUARE FOR FREQUENCY IS 5.66

SERIAL FREQUENCY FOR OCTAL POSITION 2

	0	1	2	3	4	5	6	7
C	163	167	157	147	164	166	172	150
1	170	147	159	150	147	139	150	153
2	154	143	142	158	174	162	155	170
3	161	146	154	163	134	164	160	134
4	150	159	171	145	190	138	151	174
5	166	144	166	152	150	142	147	169
6	167	164	151	149	144	166	153	158
7	155	145	158	152	175	159	164	151
	1286	1215	1258	1216	1278	1236	1252	1259

CHI-SQUARE FOR SERIAL FREQUENCY IS 49.77
CHI-SQUARE FOR FREQUENCY IS 3.85

SERIAL FREQUENCY FOR OCTAL POSITION 3

	0	1	2	3	4	5	6	7
C	147	153	151	148	152	146	155	176
1	154	176	154	160	169	183	170	151
2	145	180	154	162	157	149	141	162
3	141	166	164	125	140	128	163	165
4	160	144	167	154	143	144	153	158
5	184	155	156	149	150	149	151	149
6	169	173	134	132	154	169	169	157
7	128	170	170	162	158	175	155	172
	1228	1317	1250	1192	1223	1243	1257	1290

CHI-SQUARE FOR SERIAL FREQUENCY IS 71.05
CHI-SQUARE FOR FREQUENCY IS 8.61

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}}$

SERIAL FREQUENCY FOR OCTAL POSITION 4

	0	1	2	3	4	5	6	7
0	139	150	151	145	181	139	157	168
1	165	162	164	144	155	169	160	156
2	141	158	154	144	163	177	158	139
3	164	148	147	152	145	166	128	166
4	140	153	148	137	149	171	152	185
5	168	189	150	169	156	148	160	170
6	148	168	160	150	137	169	166	136
7	165	147	160	175	149	171	153	146
	1230	1275	1234	1216	1235	1310	1234	1266

CHI-SQUARE FOR SERIAL FREQUENCY IS 65.75
CHI-SQUARE FOR FREQUENCY IS 5.42

SERIAL FREQUENCY FOR OCTAL POSITION 5

	0	1	2	3	4	5	6	7
0	163	151	172	145	149	151	169	143
1	146	149	148	154	150	164	180	156
2	161	153	177	144	152	153	146	172
3	159	160	152	157	161	155	160	159
4	147	155	147	168	143	151	163	146
5	160	174	171	164	153	164	146	137
6	153	159	147	169	147	162	161	169
7	154	146	144	162	165	169	142	151
	1243	1247	1258	1263	1220	1269	1267	1233

CHI-SQUARE FOR SERIAL FREQUENCY IS 38.35
CHI-SQUARE FOR FREQUENCY IS 1.70

SERIAL FREQUENCY FOR OCTAL POSITION 6

	0	1	2	3	4	5	6	7
0	146	153	135	159	169	164	157	143
1	124	136	165	176	155	155	155	147
2	139	142	160	163	154	156	147	161
3	156	148	169	146	170	149	156	164
4	171	165	148	168	180	165	185	158
5	173	171	142	136	177	152	168	136
6	177	141	162	150	180	160	151	150
7	140	157	141	160	155	154	152	156
	1226	1213	1222	1258	1340	1255	1271	1215

CHI-SQUARE FOR SERIAL FREQUENCY IS 65.83
CHI-SQUARE FOR FREQUENCY IS 10.07

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}}$

SERIAL FREQUENCY FOR OCTAL POSITION 7

	0	1	2	3	4	5	6	7
0	140	168	151	151	155	148	164	148
1	170	157	158	163	173	174	134	161
2	156	163	167	137	186	162	170	155
3	139	168	166	158	145	146	154	133
4	164	151	171	154	152	149	155	177
5	168	157	167	154	140	168	143	158
6	152	168	162	140	163	152	166	132
7	136	158	154	152	159	156	149	153
	1225	1290	1296	1209	1273	1255	1235	1217

CHI-SQUARE FOR SERIAL FREQUENCY IS 53.24
CHI-SQUARE FOR FREQUENCY IS 6.31

SERIAL FREQUENCY FOR OCTAL POSITION 8

	0	1	2	3	4	5	6	7
0	152	155	168	142	148	163	148	169
1	149	152	168	157	175	141	156	178
2	153	167	153	146	167	147	149	161
3	147	168	150	148	165	146	175	145
4	145	170	150	168	146	144	163	158
5	161	156	142	164	147	150	167	147
6	170	148	148	168	149	173	144	152
7	168	160	164	151	147	170	150	152
	1245	1276	1243	1244	1244	1234	1252	1262

CHI-SQUARE FOR SERIAL FREQUENCY IS 41.11
CHI-SQUARE FOR FREQUENCY IS 0.98

SERIAL FREQUENCY FOR OCTAL POSITION 9

	0	1	2	3	4	5	6	7
0	159	156	177	158	137	175	138	157
1	157	136	176	136	155	157	156	174
2	139	155	158	156	175	158	136	178
3	156	176	157	136	176	137	155	156
4	136	176	136	157	155	157	174	154
5	158	155	157	177	157	138	175	137
6	176	156	137	173	134	157	155	156
7	176	137	157	156	156	175	155	137
	1257	1247	1255	1249	1245	1254	1244	1249

CHI-SQUARE FOR SERIAL FREQUENCY IS 77.93
CHI-SQUARE FOR FREQUENCY IS 12.96

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}}$

TOTAL SERIAL FREQUENCY

	0	1	2	3	4	5	6	7
0	1357	1402	1426	1348	1402	1431	1413	1419
1	1387	1353	1448	1395	1440	1458	1420	1417
2	1337	1429	1437	1357	1480	1431	1361	1445
3	1403	1450	1409	1355	1365	1321	1440	1394
4	1377	1414	1388	1408	1405	1363	1441	1453
5	1487	1462	1398	1435	1386	1361	1421	1366
6	1479	1435	1361	1401	1369	1476	1418	1352
7	1371	1373	1410	1438	1402	1475	1377	1368
	11198	11318	11277	11137	11249	11316	11291	11214

CHI-SQUARE FOR SERIAL FREQUENCY IS 70.16
CHI-SQUARE FOR FREQUENCY IS 2.50

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}}$

SERIAL FREQUENCY FOR OCTAL POSITION 1

	0	1	2	3	4	5	6	7
C	139	146	147	170	167	154	136	153
1	172	152	144	179	159	165	155	133
2	139	172	170	151	160	139	167	143
3	174	177	165	165	164	190	160	145
4	148	146	162	164	157	161	165	163
5	161	147	148	192	157	141	150	166
6	142	146	154	156	162	172	144	149
7	137	173	151	163	140	140	148	143
	1212	1259	1241	1340	1266	1262	1225	1195

CHI-SQUARE FOR SERIAL FREQUENCY IS 69.81
CHI-SQUARE FOR FREQUENCY IS 11.00

SERIAL FREQUENCY FOR OCTAL POSITION 2

	0	1	2	3	4	5	6	7
C	161	149	166	140	149	149	166	158
1	151	138	165	178	163	147	133	151
2	152	155	160	171	152	150	160	173
3	181	159	163	159	171	148	165	143
4	161	162	133	150	150	144	164	169
5	141	150	189	153	136	154	164	125
6	128	159	134	162	166	167	169	180
7	163	154	163	176	146	153	144	165
	1238	1226	1273	1289	1233	1212	1265	1264

CHI-SQUARE FOR SERIAL FREQUENCY IS 70.49
CHI-SQUARE FOR FREQUENCY IS 3.94

SERIAL FREQUENCY FOR OCTAL POSITION 3

	0	1	2	3	4	5	6	7
C	159	181	141	154	156	157	164	173
1	177	201	167	180	168	165	150	152
2	144	169	172	154	156	157	152	152
3	153	173	156	150	156	177	140	146
4	156	165	158	147	149	151	147	152
5	153	153	165	153	165	173	149	143
6	163	155	148	157	127	144	135	148
7	180	163	149	156	148	130	140	126
	1285	1360	1256	1251	1225	1254	1177	1192

CHI-SQUARE FOR SERIAL FREQUENCY IS 74.23
CHI-SQUARE FOR FREQUENCY IS 18.17 *

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}-31}$

SERIAL FREQUENCY FOR OCTAL POSITION 4

	0	1	2	3	4	5	6	7
C	147	180	137	148	138	153	136	149
1	151	150	144	142	160	164	172	174
2	127	155	144	152	155	158	143	173
3	141	155	153	141	191	148	176	159
4	157	136	148	199	167	138	163	167
5	177	160	149	153	155	161	133	158
6	129	149	156	165	150	162	140	178
7	159	172	176	164	159	162	166	176
	1188	1257	1207	1264	1275	1246	1229	1334

CHI-SQUARE FOR SERIAL FREQUENCY IS 86.35 *
CHI-SQUARE FOR FREQUENCY IS 11.26

SERIAL FREQUENCY FOR OCTAL POSITION 5

	0	1	2	3	4	5	6	7
C	156	137	166	163	165	149	138	157
1	154	164	147	178	154	151	160	162
2	151	170	138	152	158	150	165	156
3	158	177	168	162	158	161	162	157
4	161	152	164	173	156	133	128	178
5	166	151	162	172	152	164	144	136
6	143	160	143	140	151	170	150	143
7	142	159	152	163	151	169	153	175
	1231	1270	1240	1303	1245	1247	1200	1264

CHI-SQUARE FOR SERIAL FREQUENCY IS 52.16
CHI-SQUARE FOR FREQUENCY IS 5.12

SERIAL FREQUENCY FOR OCTAL POSITION 6

	0	1	2	3	4	5	6	7
C	168	192	160	156	159	156	181	136
1	150	155	171	158	154	144	163	139
2	164	137	166	159	147	141	182	170
3	160	156	157	165	172	128	158	152
4	157	157	157	160	163	154	141	157
5	182	140	147	148	147	147	151	143
6	168	148	161	141	176	169	176	156
7	159	149	147	161	128	166	143	145
	1308	1234	1266	1248	1246	1205	1295	1198

CHI-SQUARE FOR SERIAL FREQUENCY IS 67.87
CHI-SQUARE FOR FREQUENCY IS 8.52

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}-31}$

SERIAL FREQUENCY FOR OCTAL POSITION 7

	0	1	2	3	4	5	6	7
0	146	169	154	155	147	157	160	163
1	156	143	169	161	177	142	152	144
2	155	157	129	152	168	154	146	165
3	161	165	174	162	156	143	154	155
4	167	160	165	154	157	168	169	145
5	146	156	133	169	147	145	157	178
6	167	160	161	136	176	160	158	134
7	153	134	141	181	157	162	156	157
	1251	1244	1226	1270	1285	1231	1252	1241

CHI-SQUARE FOR SERIAL FREQUENCY IS 53.00
CHI-SQUARE FOR FREQUENCY IS 2.15

SERIAL FREQUENCY FOR OCTAL POSITION 8

	0	1	2	3	4	5	6	7
0	178	165	150	143	163	155	150	155
1	159	176	173	164	163	160	154	181
2	157	185	164	155	170	155	112	158
3	165	152	148	152	148	136	158	142
4	155	181	163	153	171	158	155	154
5	143	148	140	164	160	144	156	141
6	140	160	167	128	156	144	155	158
7	162	163	151	142	159	144	168	171
	1259	1330	1256	1201	1290	1196	1208	1260

CHI-SQUARE FOR SERIAL FREQUENCY IS 65.31
CHI-SQUARE FOR FREQUENCY IS 12.23

SERIAL FREQUENCY FOR OCTAL POSITION 9

	0	1	2	3	4	5	6	7
0	174	164	167	154	159	145	149	157
1	133	168	151	147	176	164	141	167
2	158	177	159	155	157	138	141	185
3	156	150	175	154	135	167	139	150
4	150	154	146	136	133	152	165	186
5	161	142	147	169	151	166	162	149
6	163	133	162	143	156	149	163	143
7	174	159	163	168	155	166	152	170
	1269	1247	1270	1226	1222	1247	1212	1307

CHI-SQUARE FOR SERIAL FREQUENCY IS 64.31
CHI-SQUARE FOR FREQUENCY IS 5.47

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}-31}$

SERIAL FREQUENCY FOR OCTAL POSITION 10

	0	1	2	3	4	5	6	7
C	173	150	165	177	171	146	151	171
1	147	143	177	151	162	145	156	172
2	171	174	159	155	169	136	156	142
3	179	148	158	168	145	158	169	147
4	136	171	167	152	161	146	165	161
5	154	127	157	154	141	144	146	151
6	177	160	133	183	150	150	132	143
7	167	180	146	132	160	149	153	161
	1304	1253	1262	1272	1259	1174	1228	1248

CHI-SQUARE FOR SERIAL FREQUENCY IS 72.50
CHI-SQUARE FOR FREQUENCY IS 7.92

TOTAL SERIAL FREQUENCY

	0	1	2	3	4	5	6	7
C	1601	1633	1553	1560	1574	1521	1531	1572
1	1550	1590	1608	1638	1636	1547	1536	1575
2	1518	1651	1561	1556	1592	1478	1524	1617
3	1628	1612	1617	1578	1596	1556	1581	1496
4	1548	1584	1563	1588	1564	1505	1562	1632
5	1584	1474	1537	1627	1511	1539	1512	1490
6	1520	1530	1519	1511	1570	1587	1522	1532
7	1596	1606	1539	1606	1503	1541	1523	1589
	12545	12680	12497	12664	12546	12274	12291	12503

CHI-SQUARE FOR SERIAL FREQUENCY IS 76.04
CHI-SQUARE FOR FREQUENCY IS 12.65

Serial Frequency Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}-31}$

AVERAGE	2ND MOMENT	3RD MOMENT	TOTAL RUNS	DIFF IN S.D.'S	LONGEST RUN
0.503416	0.33672	0.25308	6659	0.17394	6
0.500746	0.33310	0.24919	6716	1.17806	6
0.492954	0.32683	0.24446	6674	0.18185	6
0.501233	0.33441	0.25115	6664	0.05534	6
0.508156	0.34007	0.25517	6675	0.20557	6
0.502428	0.33531	0.25121	6708	0.98830	6
0.496034	0.32891	0.24601	6706	0.94086	6
0.497888	0.33280	0.25062	6637	0.69576	6
0.502285	0.33641	0.25367	6610	1.33618	6
0.504452	0.33802	0.25406	6712	1.08318	7
0.499096	0.33263	0.24982	6679	0.30045	6
0.501351	0.33424	0.25049	6717	1.20178	6
0.499638	0.33215	0.24857	6724	1.36781	6
0.499202	0.33358	0.25061	6640	0.62460	6
0.497473	0.33025	0.24677	6657	0.22138	6
0.500909	0.33333	0.24939	6711	1.05946	6
0.493650	0.32710	0.24449	6638	0.67204	7
0.499264	0.33237	0.24901	6741	1.77104	8
0.499202	0.33121	0.24706	6642	0.57717	6
0.497345	0.33114	0.24846	6699	0.77483	7
0.501354	0.33494	0.25171	6667	0.01581	7
0.499121	0.33328	0.25040	6661	0.12650	7
0.497717	0.33058	0.24711	6694	0.65623	7
0.500846	0.33472	0.25150	6654	0.29254	6
0.498689	0.33145	0.24786	6606	1.43106	9
0.502060	0.33583	0.25227	6695	0.67995	6
0.501654	0.33376	0.24968	6707	0.96458	7
0.498795	0.33277	0.24996	6642	0.57717	6
0.500987	0.33481	0.25141	6669	0.06325	7
0.501666	0.33444	0.25088	6701	0.82227	6
0.502848	0.33622	0.25284	6593	1.73941	6
0.497779	0.33205	0.24930	6612	1.28874	6
0.499485	0.33190	0.24805	6680	0.32416	6
0.497324	0.33106	0.24817	6577	2.11892 *	7
0.501334	0.33561	0.25235	6687	0.49020	6
0.504882	0.33740	0.25335	6655	0.26882	6
0.502317	0.33608	0.25274	6683	0.39532	6
0.501819	0.33464	0.25083	6639	0.64832	8
0.502347	0.33696	0.25384	6753	2.05567 *	7
0.502190	0.33452	0.25002	6640	0.62460	8
0.508719	0.34269	0.25873	6718	1.22550	6
0.501636	0.33476	0.25101	6629	0.88552	7
0.496322	0.33003	0.24726	6750	1.98451 *	6
0.502888	0.33514	0.25097	6611	1.31246	5
0.495526	0.32878	0.24589	6719	1.24921	7
0.503060	0.33681	0.25331	6686	0.46648	6
0.501794	0.33577	0.25237	6691	0.58508	6
0.498860	0.33251	0.24932	6617	1.17015	7
0.499773	0.33333	0.25022	6619	1.12271	7

Gorenstein's Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}}$

AVERAGE	2ND MOMENT	3RD MOMENT	TOTAL RUNS	DIFF IN S.D.'S	LONGEST RUN
0.499439	0.33339	0.25035	6699	0.77483	6
0.499730	0.33365	0.25096	6697	0.72739	6
0.492308	0.32415	0.24067	6623	1.02783	6
0.496851	0.32955	0.24620	6698	0.75111	6
0.503027	0.33663	0.25285	6619	1.12271	6
0.496214	0.33013	0.24730	6715	1.15434	6
0.504122	0.33811	0.25497	6635	0.74320	6
0.498175	0.33172	0.24852	6615	1.21759	6
0.494770	0.32910	0.24670	6677	0.25301	6
0.499750	0.33388	0.25092	6684	0.41904	8
0.501712	0.33490	0.25120	6623	1.02783	6
0.498852	0.33199	0.24872	6689	0.53764	7
0.498639	0.33257	0.24972	6636	0.71948	7
0.494429	0.32857	0.24615	6696	0.70367	6
0.503800	0.33779	0.25435	6594	1.71569	6
0.502125	0.33549	0.25203	6675	0.20557	6
0.498457	0.33252	0.24983	6699	0.77483	6
0.498404	0.33205	0.24912	6687	0.49020	6
0.499022	0.33128	0.24740	6670	0.08697	7
0.502535	0.33651	0.25336	6694	0.65623	6
0.503102	0.33627	0.25284	6674	0.18185	6
0.501612	0.33491	0.25119	6610	1.33618	7
0.495857	0.32937	0.24645	6657	0.22138	7
0.494995	0.32821	0.24516	6762	2.26914 *	6
0.495970	0.32888	0.24566	6768	2.41146 *	6
0.502487	0.33492	0.25086	6653	0.31625	7
0.502189	0.33607	0.25309	6707	0.96458	7
0.500699	0.33363	0.25005	6748	1.93707	6
0.500387	0.33353	0.25021	6731	1.53385	7
0.496065	0.32933	0.24622	6675	0.20557	5
0.499594	0.33351	0.25046	6573	2.21379 *	5
0.498094	0.33247	0.24962	6655	0.26882	6
0.500539	0.33312	0.24908	6690	0.56136	7
0.502230	0.33567	0.25244	6634	0.76692	6
0.492452	0.32612	0.24348	6720	1.27293	6
0.498236	0.33207	0.24925	6679	0.30045	7
0.503397	0.33667	0.25300	6669	0.06325	6
0.498036	0.33082	0.24731	6623	1.02783	6
0.500617	0.33514	0.25208	6676	0.22929	6
0.501164	0.33454	0.25081	6642	0.57717	7
0.501284	0.33451	0.25136	6654	0.29254	8
0.502973	0.33583	0.25202	6702	0.84599	8
0.498702	0.33133	0.24762	6683	0.39532	6
0.500668	0.33336	0.24977	6650	0.38741	7
0.497858	0.33086	0.24751	6715	1.15434	6
0.498660	0.33275	0.24982	6709	1.01202	6
0.495982	0.32883	0.24577	6710	1.03574	6
0.504976	0.33854	0.25493	6681	0.34788	7
0.496975	0.33031	0.24720	6680	0.32416	6

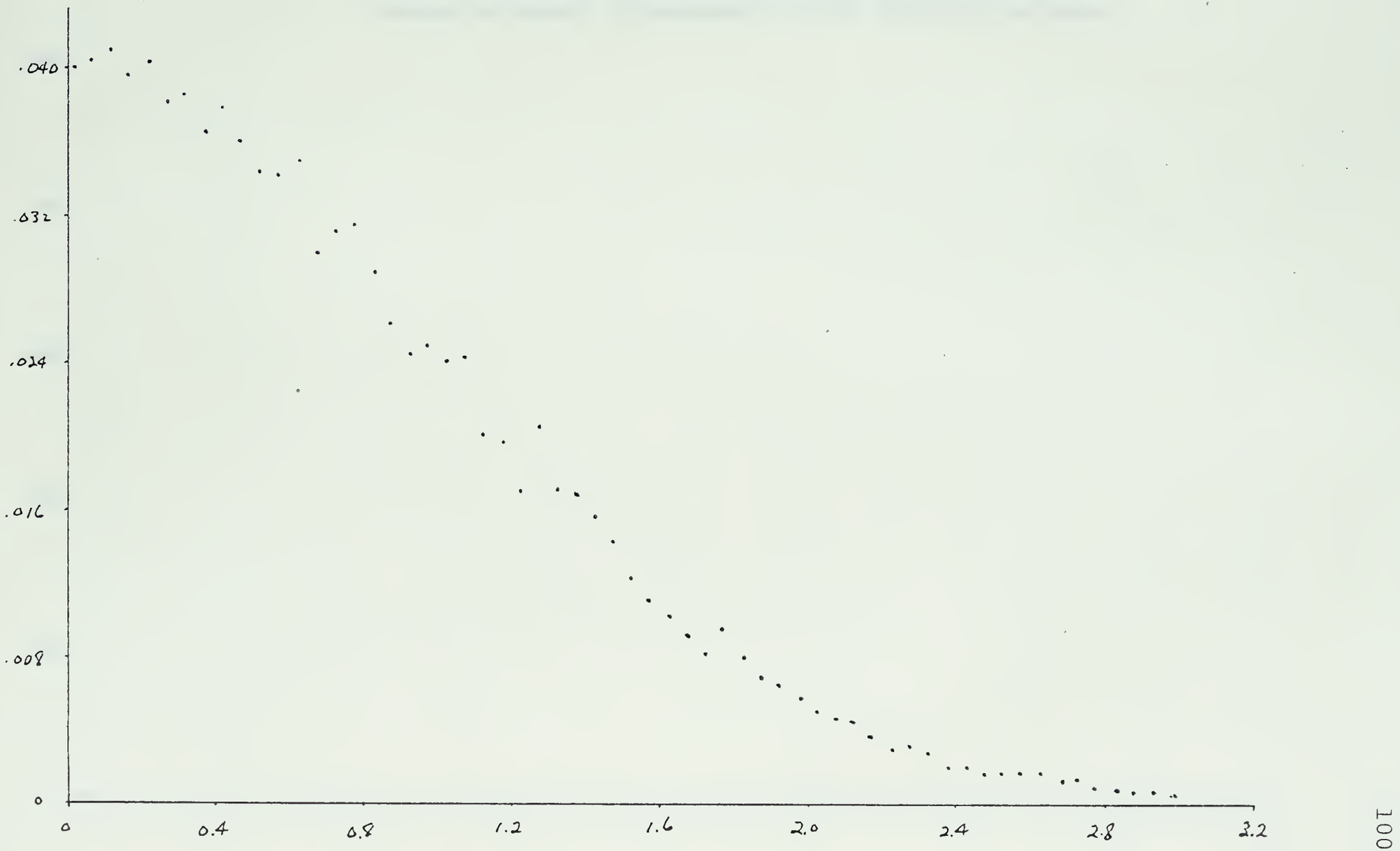
Gorenstein's Results for $x_{i+1} \equiv 5^{13} x_i \pmod{2^{35}-31}$

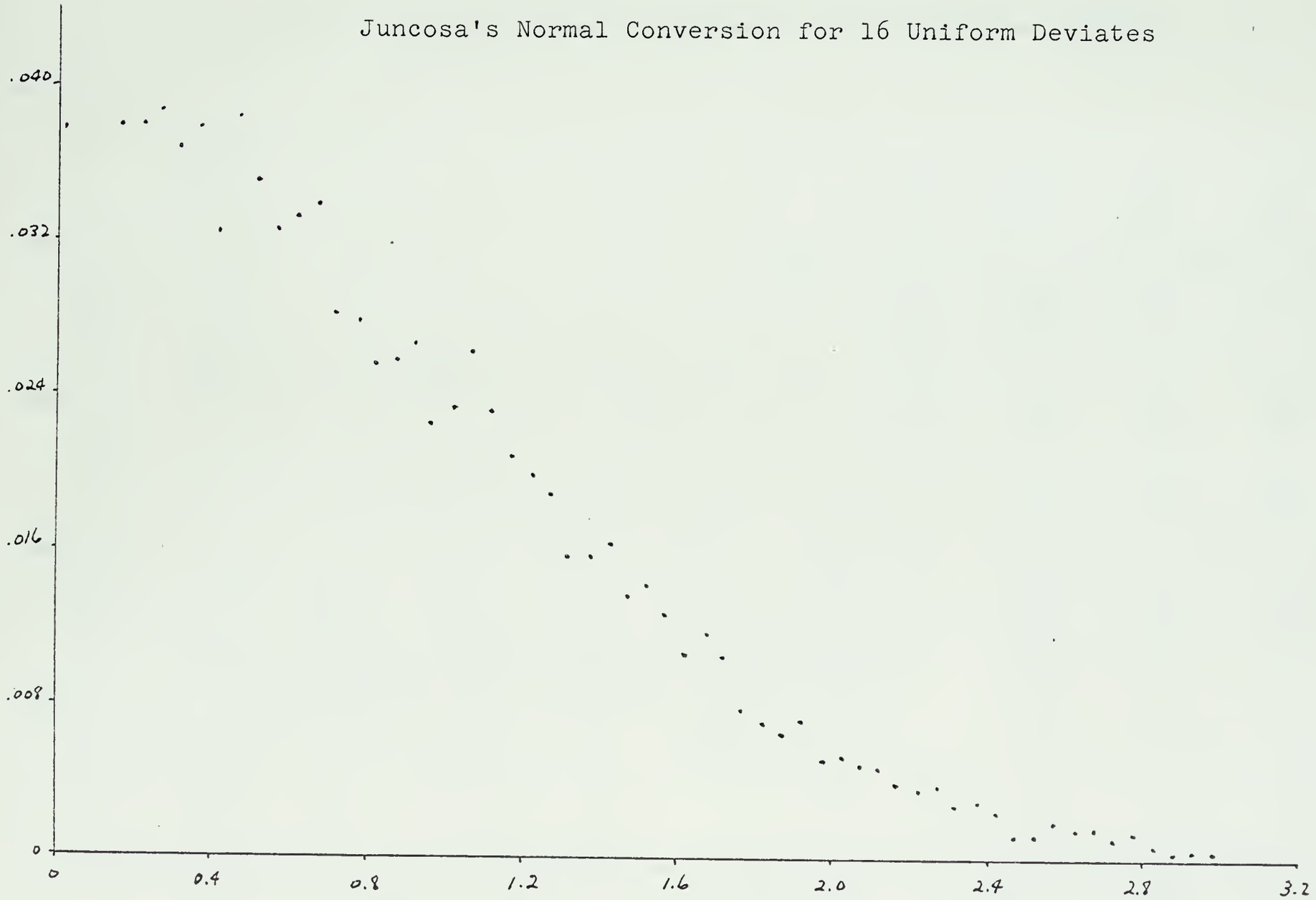
d^2	Expected Frequency	Observed Frequency	Accumulated Frequency
0.1C	0.234832	0.231560	0.231560
0.2C	0.174973	0.175800	0.407360
0.3C	C.139495	0.139000	0.546360
0.4C	0.112718	0.109160	0.655520
0.5C	0.090969	0.091640	0.747160
0.6C	0.072614	0.074880	0.822040
0.7C	0.056749	0.056400	0.878440
0.8C	0.042813	0.042080	0.920520
0.9C	0.030431	0.032280	0.952800
1.0C	0.019333	0.019680	0.972480
1.1C	0.010777	0.011920	0.984400
1.2C	0.006345	0.007480	0.991880
1.3C	0.003740	0.003720	0.995600
1.4C	0.002137	0.002440	0.998040
1.5C	0.001155	0.001280	0.999320
1.6C	0.000571	0.000520	0.999840
1.7C	0.000246	0.000160	1.000000
1.8C	0.000083	0.000000	1.000000
1.9C	0.000018	0.000000	1.000000
2.0C	0.000001	0.000000	1.000000

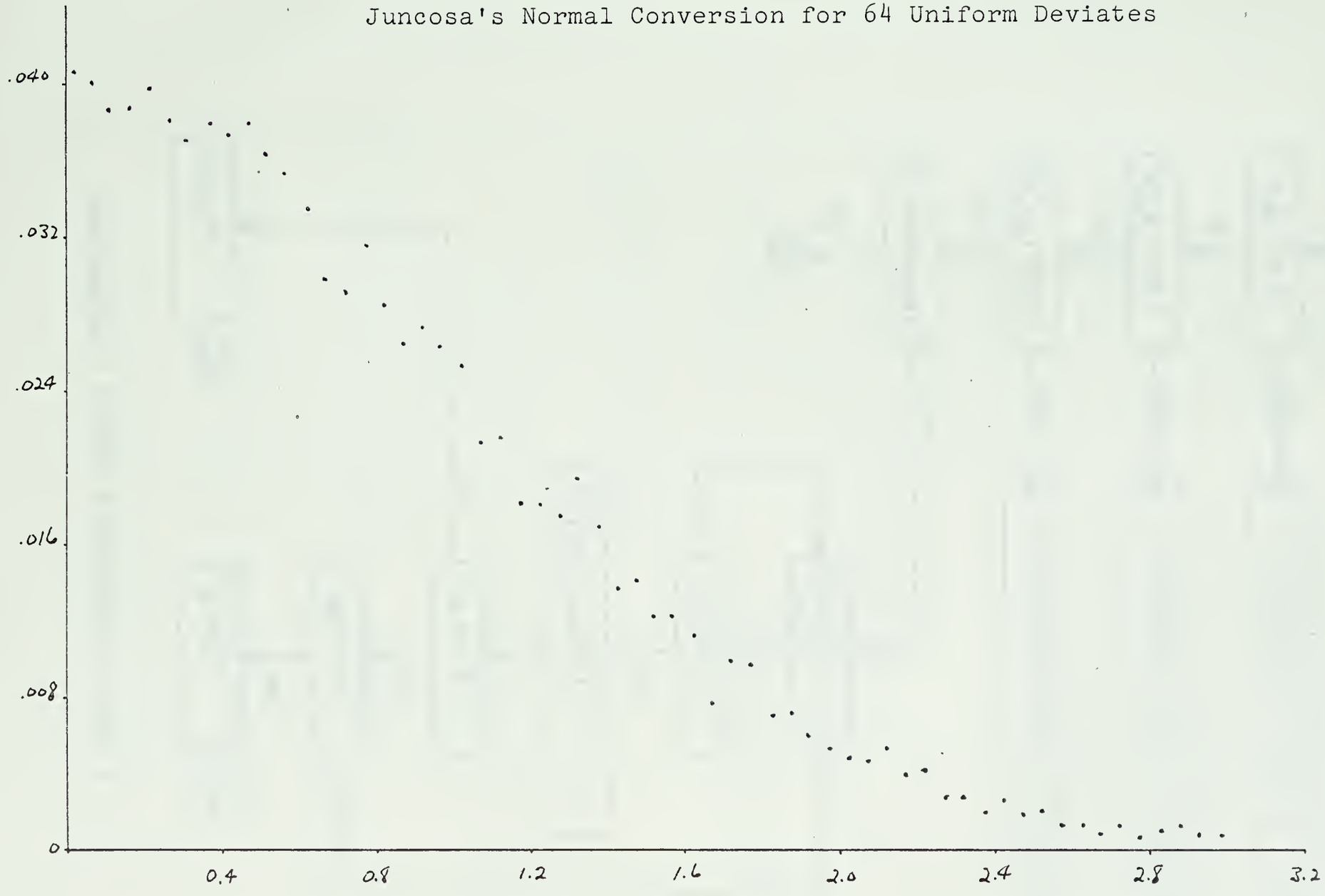
d^2 Results for $x_{i+1} \equiv 5^{13}x_i \pmod{2^{35}-31}$

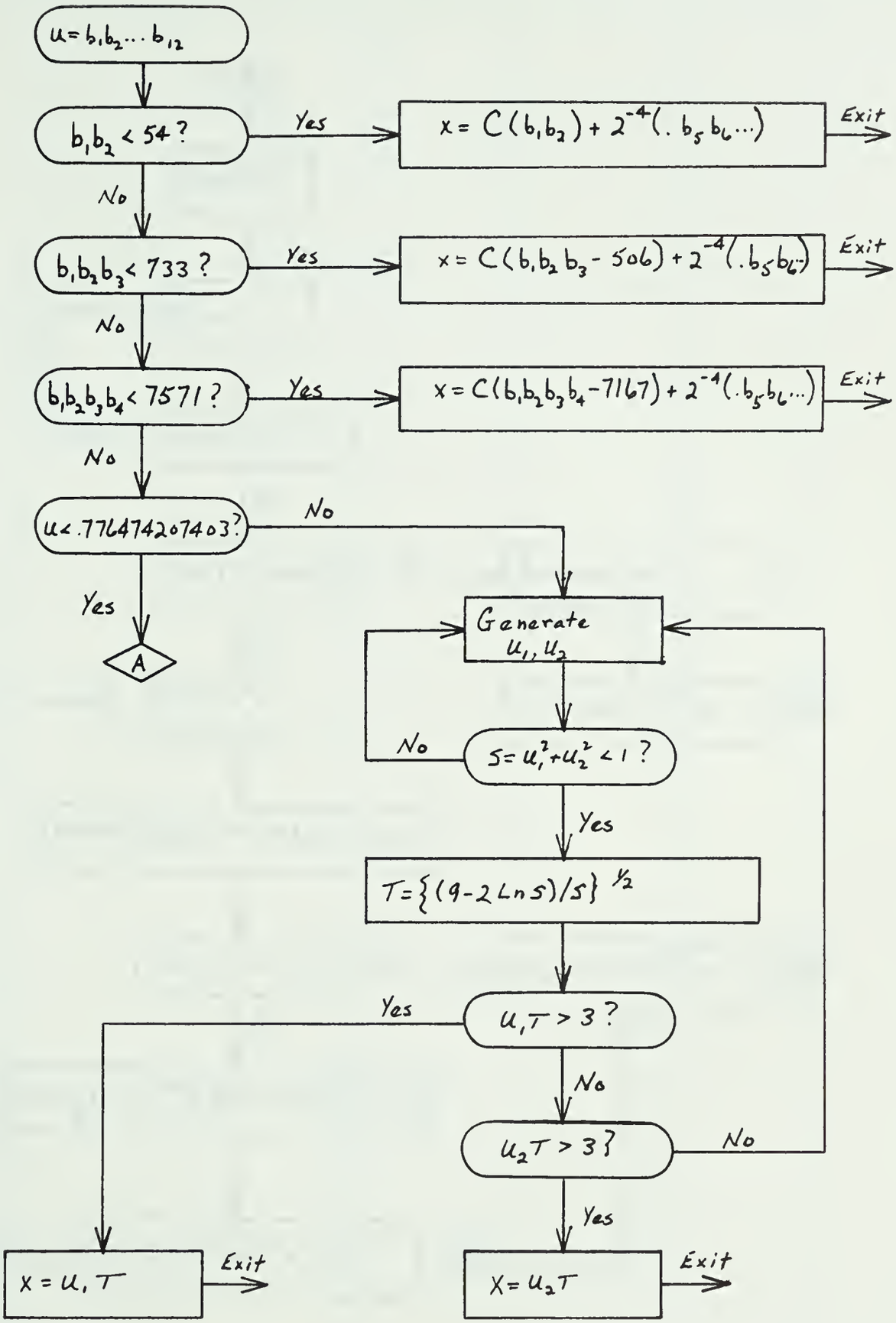
d^2	Expected Frequency	Observed Frequency	Accumulated Frequency
0.1C	0.234832	0.235160	0.235160
0.2C	0.174973	0.175080	0.410240
0.3C	0.139495	0.139880	0.550120
0.4C	0.112718	0.110960	0.661080
0.5C	0.090969	0.087400	0.748480
0.6C	0.072614	0.074600	0.823080
0.7C	0.056749	0.056920	0.880000
0.8C	0.042813	0.044960	0.924960
0.9C	0.030431	0.031120	0.956080
1.0C	0.019333	0.019440	0.975520
1.1C	0.010777	0.010440	0.985960
1.2C	0.006345	0.005920	0.991880
1.3C	0.003740	0.003760	0.995640
1.4C	0.002137	0.002280	0.997920
1.5C	0.001155	0.001200	0.999120
1.6C	0.000571	0.000640	0.999760
1.7C	0.000246	0.000200	0.999960
1.8C	0.000083	0.000040	1.000000
1.9C	0.000018	0.000000	1.000000
2.0C	0.000001	0.000000	1.000000

d^2 Results for $x_{i+1} \equiv 5^{13}x_i \pmod{2^{35}}$

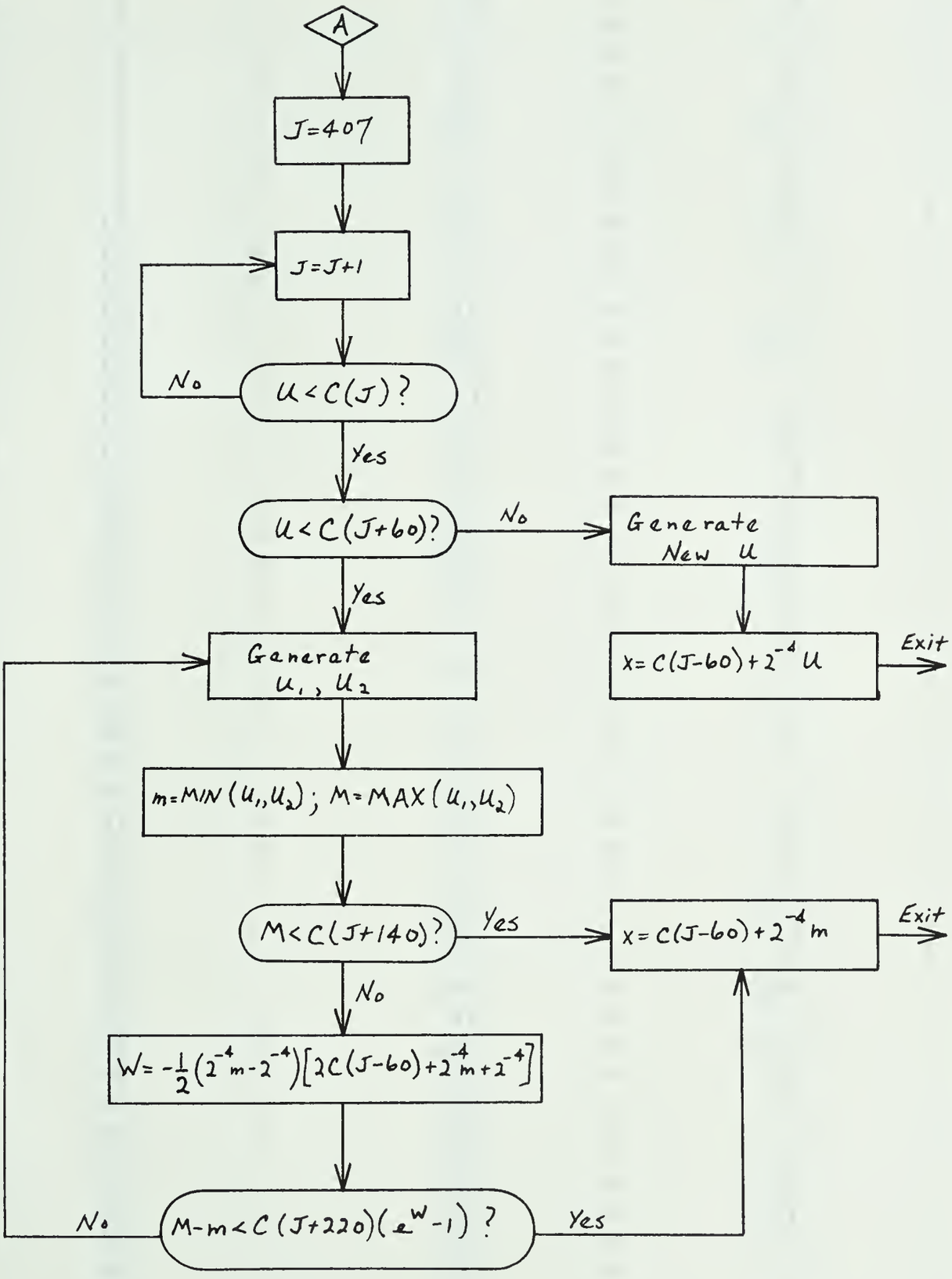

Box and Muller Normal Conversion
.040
.032
.024
.016
.008
0
0
0.4
0.8
1.2
1.6
2.0
2.4
2.8
3.2


Juncosa's Normal Conversion for 16 Uniform Deviates
.040
.032
.024
.016
.008
0
0
0.4
0.8
1.2
1.6
2.0
2.4
2.8
3.2


Juncosa's Normal Conversion for 64 Uniform Deviates
.040
.032
.024
.016
.008
0
0.4
0.8
1.2
1.6
2.0
2.4
2.8
3.2



Marsaglia's Normal Conversion - Part 1



Marsaglia's Normal Conversion - Part 2

GAME NUMBER	NUMBER OF TRIALS	GROSS WINNINGS
1	66	35
2	5	0
3	5	0
4	9	2
5	14	5
6	236	126
7	76	37
8	7	1
9	64	31
10	23	10
11	5	0
12	9	2
13	10	3
14	385	208
15	18	7
16	7	1
17	60	32
18	9	2
19	28	13
20	42	21
21	15	5
22	165	88
23	16	6
24	37	17
25	5	0
26	30	13
27	7	1
28	27	12
29	9	2
30	19	8
31	24	10
32	5	0
33	35	16
34	254	133
35	11	3
36	17	6
37	344	184
38	18	7
39	243	127
40	56	27
41	21	8
42	10	3
43	74	37
44	35	15
45	55	27
46	171	87
47	22	9
48	13	4
49	23	9
50	82	41

AVERAGE NUMBER OF TRIALS UNTIL RUIN IS 58.42

Chuck-a-Luck Results for $5

GAME NUMBER	NUMBER OF TRIALS	GROSS WINNINGS
1	26	8
2	39	15
3	298	157
4	35	13
5	24	7
6	73	33
7	78	36
8	74	35
9	117	61
10	225	117
11	183	93
12	25	8
13	94	46
14	60	27
15	18	4
16	16	3
17	30	11
18	44	19
19	128	63
20	122	60
21	435	230
22	424	225
23	70	32
24	89	43
25	57	26
26	127	63
27	37	14
28	134	65
29	104	52
30	20	5
31	126	63
32	92	43
33	26	8
34	158	80
35	12	1
36	231	124
37	279	146
38	46	19
39	186	95
40	56	25
41	209	109
42	184	95
43	127	63
44	157	80
45	61	28
46	76	37
47	15	3
48	49	20
49	465	247
50	232	119

AVERAGE NUMBER OF TRIALS UNTIL RUIN IS 119.86

Chuck-a-Luck Results for $10

GAME NUMBER	NUMBER OF TRIALS	GROSS WINNINGS
1	138	63
2	430	220
3	126	58
4	96	41
5	83	34
6	267	137
7	2057	1107
8	123	55
9	114	51
10	94	40
11	416	211
12	132	60
13	356	187
14	217	105
15	543	289
16	145	68
17	466	243
18	75	31
19	208	99
20	42	12
21	64	24
22	181	86
23	271	137
24	195	95
25	163	76
26	266	130
27	250	125
28	320	161
29	141	66
30	203	99
31	111	49
32	130	58
33	100	43
34	106	47
35	609	324
36	129	60
37	248	123
38	151	73
39	240	119
40	166	79
41	68	26
42	243	122
43	211	103
44	432	222
45	82	34
46	139	64
47	298	150
48	215	101
49	48	15
50	95	39

AVERAGE NUMBER OF TRIALS UNTIL RUIN IS 240.06

Chuck-a-Luck Results for $20

Throws	Crosses	Area
1C000	3334	0.3334C000
2CC00	6651	0.33255000
3C000	9980	0.33266667
4CCCC	13272	0.3318CC00
5C000	16595	0.3319C000
6CCCC	19897	0.33161667
7C00C	23193	0.33132857
8CCCC	26552	0.3319CC00
9CC00	29836	0.33151111
1CCC00	33176	0.33176000

Results for $\int_0^1 x^2\,dx$ Using Buffon's Technique

Throws	Crosses	Area
1C000	2463	0.2463C000
2CCCC	4956	0.2478CC00
3C000	7452	0.2484C000
4CCCC	9933	0.24832500
5C000	12420	0.2484CC00
6CCCC	14818	0.24696667
7C000	17279	0.24684286
8CCCC	19845	0.24806250
9CC00	22310	0.24788889
1CCCCC	24806	0.24806000

Results for $\int_0^1 x^3\,dx$ Using Buffon's Technique